Discrete-Event Simulation Data Transformation:

A Model-Driven Data Integration Approach

Jamiu Mohammed Yusuf

A Thesis Submitted in Partial Fulfillment of the Requirement

of De Montfort University for the Degree of Master of Philosophy

June 2016

De Montfort University

Declaration

I declare that the work described within this thesis was originally undertaken by me, (Jamiu Yusuf) between the dates of registration for the degree of Master of Philosophy at De Montfort University, April 2014 to June 2016.

Acknowledgments

First, I thank God for giving me the strength and inspirations throughout this two-year journey.

My deepest appreciation is extended to my family (Yemi Khadijat Jamiu, Haffeez Gbolahon Jamiu and Hanifah Olatanwa Jamiu) for their endless support and love.

I would like to express my appreciation to my supervisors Dr. Seng Chong and Dr. Parminder Kang for their valuable supervision and constructive suggestions during the planning and development of this research. Their willingness to give their time generously has been much appreciated.

I extend my warm thanks to the great people at the Lean Engineering Research Group, both academics and students, for the help, guidance and support they have given me. Wish you all the best of luck in your life and career.

Finally, I thank De Montfort University for the opportunity to undertake this research and the Graduate School Office for their help in smoothly administrating my path through the administrative and training processes.

Dedication

To my parents and my wife, Yemi Khadijat Jamiu and My children, Haffez Gbolahon Jamiu and Hanifah Olatanwa Jamiu.

Abstract

Achieving a smooth production system is a complex process that requires the use of commercial discrete event simulation (DES) tools to provide a high flexibility production process, for instance the use of simulation modelling to model a production system. These tools require high levels of cooperation to work together because they are not designed to be integrated and hardly share their data. This research aims to integrate DES tools applied by different manufacturing systems in order to enable them to share their data.

This thesis presents data integration from a simulation model point of view because it views data integration between different DES tools models as key steps towards system integration. A new approach has been developed which is called a Model-Driven Data Integration Approach (MDDI), so named because the integration involves the combination of data from different DES tools model sources.

The effectiveness of this data integration approach has been demonstrated in a case study undertaken for DES design of a phone production line in the manufacturing industry. However, the application of the MDDI is not limited to this case study: it can also be used for other system and applications.

The MDDI approach was tested and evaluated on the basis of this case study. These test cases simulated how the data integration based on different DES tools' models react to the process of data sharing as they occur in the manufacturing production line. The result is that the MDDI approach best maintains data consistency and integrity and can be adopted by different industries.

Table of Contents

List of Figures

List of Tables

Glossary of Acronyms

Acronym	Definition
IBM	International Business Machines Corporation
GAV	Global-as-view
XML	Extensible Markup Language
ME	Modelling element
DES	Discrete Event simulation packages
VIM	Visual interactive modeling
VBA	Visual Basic Application
NIST	Institute of National standard technology
RDFS	Resource Description Framework Schema
CAEX	Computer Aided Engineering Exchange
CAD	Computer-aided design
DB	Data Base
UML	Unified Modeling Language
ATL	Transformation language
MDDI	Model Driven Data Integration
SPFM	Simul8 Process Framework Metamodel
APFM	ARENA Process Framework Metamodel (APFM)
XMI	Metadata Interchange
OMG	Object management Group specification

Chapter 1 - Introduction

1.1 Introduction

An increase in global competition has forced industries, such as manufacturing systems, to adopt various techniques such as simulation modelling in order to keep pace with the highly complex nature of production and advancement of technology and achieve their main goals of on-time product delivery and lead time reduction. The demand of customers for both quality services and price reduction has made the manufacturing process more complex than expected in the past few decades. One of the tools adopted by the manufacturing companies to handle this complexity is the DES that allows them to model every aspect of the production and decision support in order to cope with these fast track market changes. The DES is a commercial software tool regarded as a dominant one that provides a large amount of flexibility for analysis and modelling of a system widely adopted and used in many areas such as military, health finance, human resources, manufacturing and energy. The DES has had a consistent impact on production planning, resource allocation and strategic planning (Skoogh, et al., 2012).

One of the challenges that have hindered the smooth usage of the DES is the issue of data sharing among its different types, with the models heavily relies on its data to estimate various parameters to use and drive the models through simulation and time (Law, 2007). Data is one of the most vital aspects of any system as it plays a key role in informing the decision-making process, increasing performance, linking information, and allowing for an effective strategy (Davenport & Harris, 2007). For instance, organisations use data for managing performance, planning, and trend analysis. Consequently, the future of any organisation relies on its data, with policy- and decision-makers using it to make decisions that will improve their operational efficiency.

Faced with global economic challenges, business processes (e.g. The manufacturing process) have employed simulation software tools to model their complex operation in order to increase efficiency and get their products to market earlier. A typical simulation tool runs differently and expresses its data independent of other tools, therefore causing the company to struggle to meet the high expectations of the competitive market. For example, commercial discrete event simulation (DES) tools express their data in different formats, and these DES tools should share their data with others, but while transforming and integrating them their results are subject to various forms of language difference, format issues, and semantic heterogeneity (Wan-teh, et al., 2007).
‘Semantic heterogeneity’ refers to data sets for the same system that are produced by different vendors, resulting in different data representation and meaning (Bergman, 2006). Beyond the issue of meaning and representation, data semantic heterogeneity is compounded due to the several classification methods applied to data sources. Yet, for DES sources to share data with one another there is a vital need to address this semantic heterogeneity (Bergman, 2014).

Presently, data continues to experience rapid changes as more simulation tools make their way to market. Moreover, these data sources are not intended to share their data with other tools. Therefore, transforming and integrating entities from one tool with entities coming from other tools has always resulted in different varieties of heterogeneity (Wan-the et al., 2007).

According to Zeigle et al. (2000), ‘entities’ are units of data in a simulation tool that can be classified by their relationship to other elements. Some examples of entities that relate to this research are work entry point, work centre, resources, work exit point, create, etc. More details are given in Section 3.3.

Data sharing can be complex, with DES tools being customised by the producers (Bergman, 2006); this has prompted the software industry to adopt a different view of the data transformation process: for example, the data tools federation (Bruni et al., 2003) uses a database method as a means of providing access to data from different sources.

Essentially, the data federation method is an example of tools being developed to exchange data from different sources, which then provides the end users with the unified database; however, this lack both the credentials and flexibility to address the issues of data sharing among DES (Bruni et al., 2003).

A grid computing tool (Calvanese et al., 2005), is another method that provides a useful framework for data vendors to cope with the heterogeneous nature of the various data sources in the simulation system; however, this tool only examines the data point of view without accommodating the modelling elements of the models and consequently cannot accommodate DES data integration needs.

In an attempt to solve the DES semantic heterogeneity, researchers have developed solutions such as schema integration (Amit et al., 2005), and data interoperability (Bergman, 2006); however, data sharing is not yet fully achieved (Bergman, 2014). Therefore, for the multiple DES data sources to share data with one another, there is a pressing need for more research in this area.

The focus of this research is to develop a flexible transformation and integration approach that will allow data sharing among different DES. To achieve this, this research initially analysed different DES data models in order to determine how to access their data sources.

1.2 Data Sharing

The sharing of data across different DES involves different scenarios, such as (i) data transformation and (ii) data integration; these two scenarios involve other processes such as mapping, concept definitions, and data class definitions, as described in Chapter 3.The data transformation scenario is the first step in which rules are defined for transforming a source model and its associated data and this is defined according to the structure of the target data; consequently the two data sources concept for modelling is defined.

1.3 Problem Description

The manufacturing industry has invested substantial resources into creating discrete event simulation (DES) to model their complex production systems, and are required to spend higher resources to research how the DES data can be shared and reused among the different packages they use (Jianbo et al., 2015). Data sharing has been a major problem for many system domains, for example, DES e.t.c. In particular, the ability of a simulation tool data to be ready for use by another system and to also allow the users to have a unified data structure has been difficult to achieve (Lenzerini, 2002). The data sharing is relevant to many applications that are not limited to business, e-commerce applications, and enterprise information integration and simulation systems.

Data sources have to be transformed and integrated to allow tools (e.g. DES) to consume data from other sources or among themselves, but DES have adopted different ways of presenting their data and using different representations which made their reuse complicated. The behavior of the producers of these tools to represent their data in different forms presents challenging problems (e.g. Heterogeneity).The heterogeneity in a DES tool data can be categorised as schematic, semantic, and syntactic (Bishr, 2008).

Schematic heterogeneity describes types of heterogeneity resulting from organisational differences, for example generalisation and aggregation. Syntactic is a type of heterogeneity that is a result of language and the use of different model concepts, for example XML and relational data.

Solving semantic heterogeneity issues will reduce the time and effort spent by the users of these tools to integrate their data, especially during the manufacturing process. Therefore, to this end, there is a need for new solutions that can allow the DES tools to share their and data and make the development of simulations cost-effective.

1.4 Research Gap

Several techniques for data transformation and integration have been developed. Techniques such as that of (Hyeonsook, et al., 2009) have handled different case studies in data sharing of semantically heterogeneous data sources, but failed to accommodate the problem of the model data transformation and integration for DES tools.

Data sharing in this research is the ability to allow one DES simulation data to be integrated and ready to be consumed by a different DES tool. The identification of a suitable technique for data transformation and integration can accelerate the time to market for products and reduce the cost of data integration among the manufacturing production system and also provide a platform on which data can be shared among various tool sources.

Still, there are more problems, such as nomenclature, semantics, and heterogeneity issues to solve, as existing methods are far from meeting the needs of heterogeneous discrete event simulation tools, while on the other hand, the existing solutions are still far from being efficient (Jianbo et al. 2015).

1.5 Scope

This research aims at developing a flexible approach for the transformation and integration of discrete event simulation (DES) tools' data sources.

The scope of this research is limited to commercial DES types of simulation software. Therefore, this research considers two cases (ARENA and SIMUL8) model data to validate manufacturing production line data. These two models are part of the DES tools and have been extensively used for modeling and simulating the manufacturing process.

Others DES packages such as SIMIO, AnyLOGIC, ProMODEL, etc. are not included in the scope of this research and therefore will not be included. The consideration is drawn from the context of using DES tool to run manufacturing system and generate data that represent two model data and consequently, analyse and transform it so that they can consume back what is transformed.

In terms of this research toward the achieving of the data integration, it examines the modelling elements and their associated attributes as applied to tools under investigation; this represents the data model of the simulation tools and subsequently, manufacturing data are applied to develop the simulation system to help gather the required data in enhancing the transformation and integration purpose. There are many challenges faced by the DES end-users, such as manufacturing systems that relate to data integration and how this tool can cope with manufacturing complex system. The challenges that are associated with this research are (1) data representation and (2) data sharing (naming: nomenclature) or which format will the data have, this challenge is posed by the desire of the producer of these tools to customise them for profit without a concern for the end-users.

In term of manufacturing data, this research focuses on processes involved in producing a product (phone part) in a manufacturing production line, which entails that each production involve entities such as the modelling elements (ME), ME name, queue, resources, distribution of resources, time to completion of a product, inter arrival time, average time in the system etc. This research will increase the desire and interest among researchers in the field of simulation system, data management and any system to adopt different ways of representing their data.

1.6 Research Aim and Objectives

It has been a practice in the modern manufacturing industry to adopt the use of DES tools to model and simulate their complex production processes in order to achieve a smooth production system and cost effective, quality and on-time delivery of products to their customers. The use of these tools has drastically helped the manufacturing industry to reduce the cost of production, but while the industry invests a substantial amount to acquire the DES tools at a high price, they are also faced with the issue of its data reuse which makes them invest even more in the integration of their data

Therefore, the basic aim of this research is to develop a flexible data transformation and integration approach for data sharing among DES tools that takes account of manufacturing industry data sources.

To accomplish the main aim and to address the research gap, this research outlines the following objectives.

Objectives

1. Development of two models representing the DES data models that take consideration of manufacturing production line data sources.
2. Generation of a generic representation of relationships among DES data sources with reference to production line and use of DES modelling element and other attributes to obtain the model data.

3. To identify, through the literature, the concept and language definition and process interaction in different DES (e.g. ARENA and SIMUL8).
4. To establish relationships between different DES through their modelling elements identified in Objective 1 and their process interactions.
5. To develop mapping by using the relationship between the DES tools to develop the Model Driven Data Integration (MDDI) technique that can enhance data sharing among DES data sources.

The particular interest in this research is the Manufacturing Industry aspect of using the DES tools; however, there is no limit to the process and applications for non-manufacturing processes that also uses the DES tools, especially other environments where a complex system is used. For example, the health sector, military applications, aviation, finance and administration etc.

The success criteria of this research are that it has been used for a Manufacturing Industry case study. Therefore, with the valid assumption that any environment that requires the same data integration process has a set of data arising from using the DES model which must comply with the objective of this research, then it can adopt this method. This research will play a vital role in many industries as highlighted above, although it is worth bearing in mind that integrating DES tools itself is complex, therefore, the user can only ensure that it maintains the consistency and integrity of its data.

1.7 Research Methodology and Process

Research methodology can be defined as a way of carrying out a research investigation. Similarly, research methods are techniques used in research to help determine what can be done to solve the research problems. Walle (2005) described three types of research methodology, namely: quantitative, qualitative, mixed methods and multi-methods. In this research only the qualitative is unsuitable; rather, the multi-method qualitative approach shares the advantage of allowing different methods in a single step research.

The general formulation and how the researcher sets out to answer their research question are called the research strategy (Saunders et al. 2009). As for this research, "e*xperiment"* was used for the implementation of a data transformation technique that allows data sharing between different DES tools. This transformation was undertaken in Eclipse transformation environment and a manufacturing production line data was used to validate and examine the transformation and integration of the tools.

1.7.1 Research Process

To achieve the research objectives, this research adopts a multi-method quantitative method; therefore, the initial data were collected through literature review and development of DES models at conceptual level using their artificial data. Alongside this, modelling elements representing the real life process of DES was adopted for the data collections and this enables quantification of the DES application process. The research methodology in this research involves two DES as described in section 3.1.3. The discrete event simulation (DES) tool has been adopted in this research to represent the working principle of the model representation, interaction and concept definition. Literature has also shown that there a benefit in using the DES (Sandanayake et al., 2008):

1. It allows for the measurement of a discrete sequence of events with respect to time base on the basic modelling element and attribute of the models. It also allows for the investigation of a problem complex system.
2. It allows for the measurement and the illustration of complex systems such as different internal model definitions. However, this research is examining the five basic modelling elements as presented in the DES model.

The process of transformation and integration process increases the cooperation among different DES tools that incorporate a manufacturing production system life cycle to enhance manufacturing data reusability and reduce the lead time and time to market of products. Suitable data are important for different DES tools and for subsequent use of the manufacturing production process which is affected by choosing the appropriate and suitable experiment methodology, therefore, the main experiment developments are:

1. *Phase 1*: Full development of DES models to have the in-depth knowledge on how it is modeled and used for difference processes and systems, for example the manufacturing production line process.The DES models helps in collections of the following information: (i) DES data concerning the different DES tools (ii) process interaction and concept as define using the relationship between the DES tools (iii) Modeling elements and attribute concerning the DES tools (iv) resources, time, queue and distributions as applied to production line and DES data (v) total simulation time used in collecting the required data.
2. *Phase 2:* The second phase of the development entailed further data collection through the following: Collection of data through the development of the DES model and identification of its Modelling elements and their associated attributes, this time on the Manufacturing production line involving a phone part product data, namely; items, resources, arrival time, type of productions, processing time (see details in section 4.4). The criteria used in choosing these data collection methods included (i) availability of information concerning both DES tools and the manufacturing production line (ii) existence of the relationship between the DES tools and their concept definition framework (iii) the available information at all stages of the production process (iv) level of resources and processing time, and (v) length of time required to collect the data.

1.8 Chapters outline

Table 1.1: Outline of Chapters

Chapter 1: This chapter introduces the research, scope, aim and objectives, and research gap.
Chapter 2: In this chapter, a general overview of the research is presented. The overview is comprised of related literature and examples of existing discrete event simulation packages, their model representation, modelling element and their attributes.
Chapter 3: This chapter elaborates in more detail on discrete event simulation data transformation and integration. A more detailed review is also presented in this chapter, including the barriers and issues with data sharing among the available DES. The review in this chapter results in the choosing of the methodology adopted in this research.
Chapter 4: The overall methodology chosen for this research is presented in this chapter. This chapter also covers the steps of the research method as well as the data collection process
Chapter 5: The results and the analysis of the initial experiment are presented in this chapter.
Chapter 6: The conclusion, future work, and research summary for the MPhil report are presented in this chapter.

Chapter 2 - Overview of Data Transformation and Integration in Discrete-Event Simulation Packages

2.1 Introduction

Simulation software tools have been the subject of research in the past few decades, including its integration (Bergert, 2007) model data sharing, (Douglas, 2006) cooperation and their reusability (Bengtsson et al., 2009).

The integration aspect of data sharing has been the major area of research, especially concerning the simulation software in a range of uses by different application and systems such as manufacturing. The most significant area of interest to researchers has been how these DES models can share their data (Bengtsson et al., 2009) and enhance cooperation among the different simulation systems.

This chapter begins by examining the general view on the current status of integration from the general perspective of simulation system cooperation as well as looking at the specific implementation of simulation software integration. Therefore, since the aim of this research is integration that allows simulation tools to share their data and improve simulation reusability, this literature will also examine the current requirements and the problems as well as the existing methods to solve the data sharing problem among the simulation tools.

2.2 Cooperation for Different Simulation System

Determining how simulation systems cooperate is the starting point for this literature review. To have a common data representation and better interoperability among the simulation systems, efficient cooperation with one another is essential for a competent integration process (Hao et al., 2006). This research evaluates some of the problems and fundamental requirements for the simulation systems' cooperation and also examines the existing methods to improve the cooperation.

2.2.1 Problem and Fundamental Requirements

Even though the simulation system is built for the same domain and applications by independent party or producers, the actual cooperation to share data is less developed. The most prevalent challenges are the lack of cooperation amongst the producers of these packages to share their data. Researchers have not yet defined the standard method for transformation and integration that will enable simulation systems to share their data portably and effectively (Douglas, 2006).

The requirement for ensuring that data are shared among simulations becomes a very crucial topic of the simulation software as well as many applications and tools (Hyeonsook et al., 2009). The problems of data sharing are the lack of governing the standard transformation and integration method that enables the system to work together (Bengtsson et al., 2009).

Existing studies and researchers have not considered the Simulation data sources and internal data structure, but instead studies have only focused on generalising data integration and data sharing in applications that cannot be compared to simulation system and their models (Siebers et al., 2010). Therefore, there is a need for more research towards achieving automatic data sharing and interdisciplinary among simulation systems (Bengtsson et al., 2009).

2.2.2 Existing Methods to Improve the Cooperation

To improve cooperation in simulation system processes, many researchers have been motivated by different concepts to see how they can effectively collaborate effort to allow seamless interoperability among the simulation systems. Approaches by Simeone et al.

(2011) proposed a collaborative project that can be effectively used to realise cooperation among different software. Similarly, the research provides an innovative research environment to support and improve multiple collaborative integration across multiple simulation tools. He developed infrastructure for sharing called infrastructure repository. Kondylakis & Plexousakis, (2011) also applied an ontology-based data integration approach to increase collaboration among systems.

2.3 Existing Strategies for Data Integration

As it applies to data integration, the form in which the data source is represented will always determine the type of approach to be used. Specifically, when a data source is defined by the mapping of each element in the model, then the method is termed ‘Model-Driven data integration approach’ (Hyeonsook et al., 2009). To improve data integration among software packages, several methods such as (Hyeonsook, et al., 2009) and (Bengtsson, et al., 2009) have been developed and adopted in the past decade. Their research proposes a very good integration approach to model, interchange and transformation that can be proactively applied to achieve data utilisation, interoperability and portability across tools using standard development method.

Similarly, Macura (2014), targets the implementation and the creation of a new architectural platform for transformation and integration of data across different heterogeneous sources and systems. He proposed data integration from heterogeneous sources using the Extract, Transform and Load (ETL) Technology to support the implementation of knowledge discovery in operational data.

In the field of simulation systems, researchers such as Youcef and Abdelhabib (2012) established ontology for automatic code generation in Extensible Markup Language (XML) format and provided a transformation for interoperability between different systems.

Other approaches, for example Maurizio (2002), pursued the concepts of modelling and data integration and data sharing between different tools; he recommends a theoretical framework to enhance collaboration between the applications.

2.4 Problem of System Integration and Data Sharing

The problem of the integration and sharing of data is the challenge posed by simulation systems that do not share a common understanding and work together flawlessly. Different models built from different simulation systems solve different kind of problems but are applied differently and independent of each other. The solution that will allow the sharing to happen flawlessly has not yet been achieved (Bergman, 2014), with researchers focusing more on describing each systems' models without considering how the commercial simulation system can share their data by providing an approach that can solve issues such as the data format, language and semantic heterogeneity as contained within each simulation system.

2.4.1 Heterogeneous Simulation System

Simulation software tool is a useful tool that is widely applied for modelling and analysis across various disciplines that are not limited to healthcare, military applications, academia, manufacturing, etc. (Skoogh et al., 2012). The simulation tools rely on their data to drive their model in order to estimate different parameters but have only been designed to complete the task without considering the end users that will need to integrate their operational data to be shared among the different tools that are key drivers to proper decision making and allow them to have a competitive advantage over their rivals (Davenport & Harris, 2007). The integration platform to address this issue for simulation tools is still lacking (Bengtsson et al., 2009). The main reason behind this deficiency is that the producer of these tools are only concerned with achieving their set requirement for each simulation they produce without any concern as to how these tools will be integrated (Drath, 2010).

The available integration platform often contained a predefined set of processes and the line the integration will follow with the homogenous data model, this often works according to the defined motive of the integration tool which cannot be extended to incorporate other simulation tools (Moser T et al., 2010).

Some of the problems associated with the integration tool for lack of seamless platform to incorporate the requirement of other tools can also be associated with the high cost of license and other problems like the tool requirements which are independent to others (Drath, 2010).

2.4.2 Incompatible Terminology

For any system to interchange its data, ME and their attributes are the key issues that need to be resolved due to terminology differences. The reason that each model uses different terminology is that it suits their usage and concepts, leading to customisation of models (Carlos, et al., 2013). Various misunderstandings and interpretations of terminology for data sharing have arisen due to problems related to the individual application of terminology (Lafortune & Cassandras, 2009).

2.5 Data Sharing Methods by Means of Standardized Interfaces

1. ***Ontology:***

Ontologies are another critical aspect of data sharing; the ontology is a philosophy defining the XML heterogeneous sources (Figure 2.1).

XML ontology considers settling structures that are communicated in a Resource Description Framework Schema (RDFS) in order to empower semantic interrelation within the XML sources (Akella, et al., 2005).

The ontology is a mix procedure structure for XML sources that allows for the effective modification of modelling with applications. This method is useful, and because of its characterisation by extra metadata that enables the encoding of XML schema system for mapping, it is therefore helpful in this research (Eric, 2008).

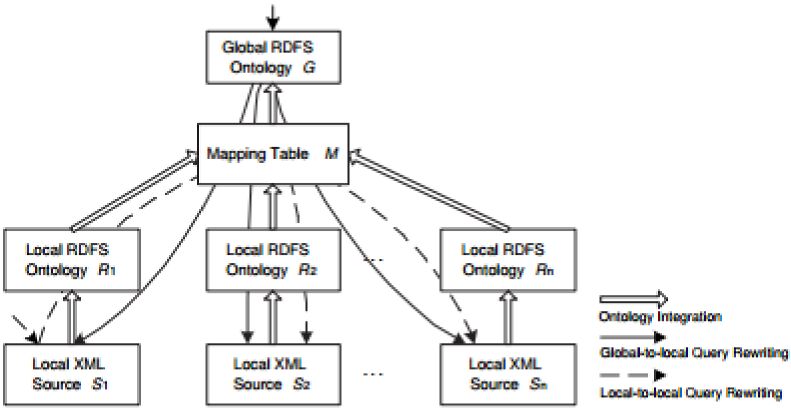


Figure 2.1: Ontology-based framework for XML data source integration (Akella, et al., 2005).

Ontology establishes a common semantic for modelling and has not really been fully developed due to application customisation by individual companies to define its wording, language, and own granularity understanding and structures (Virginija & Rimantas, 2011):(Huiyong & Isabel, 2004).

Problems of ontology

Ontology establishes a common semantics for data modeling, but lacks a model driven interface that is needed to facilitate the requirement of the integration process and requires a solution for the proper transformation of incompatible heterogeneous data sources (Lina & Robertas, 2009).

2. ***Computer-aided METK (Manufacturing Engineering Toolkit):***

METK is a method developed by the NIST. The toolkit consists of the commercial software application (COTS) used in manufacturing companies' computer workstations. The aim of the toolkit is to provide a common database, an integrated framework, and a standard interface for software applications used in manufacturing engineering (Michael, 2009).

The METK involved two processes: the first process is to collect a list of data from different tools, which is a list of elements of data sets, and the second is the validation of the data sets involved, which is the first step in this research. The METK method will contribute to the current research.

Problems with Computer-aided METK

The METK method identified some of the problems (e.g. Data format, semantics, etc.) in data sharing between commercial simulation software, such as the fact that input data from a manufacturing software application must be able to be compatible with other applications. The issue of data format is another area that needs to be automated, so that if a software application produces and generates a data set in a particular format, it should be readable by other software systems. This problem of data format and semantic heterogeneity will be addressed in this research using the model driven integration technique that accommodates semantics of any tool.

3. ***NIST SIMA (Systems Integration of Manufacturing Applications):***

The NIST undertook the program of System Integration for Manufacturing Application (SIMA) as a major effort towards achieving intra-natural (data) sharing in the manufacturing domain. The initial development started in the year 1994 with the aim of working with other industries to achieve seamless integration of various kinds of product data (Bloom, 2004).

The major achievement in this method is the development of interfaces and exchange standard protocols for manufacturing data sharing problems and transfer of data to other manufacturing enterprises.

The problem with this method is that it cannot tackle every existing data sharing problem as a result of multiple dynamic data sources of simulation software packages.

4. *CAEX (Computer Aided Engineering Exchange) format method:*

CAEX is used for hierarchical model information transformation between different manufacturing simulation tools. CAEX and XML format have some similarities in data exchange, with the benefit that they can be processed easily due to inbuilt support for XML and compatible queries.

Miriam and Rainer (2008) described CAEX as a static method that requires a protocol to enable it to read the files before transforming to another file; they described it as not flexible enough to handle DES heterogeneous data. Therefore, this research will address this problem by a direct mapping of the data from one DES to another.

5. *Scenarios Navigator-Based Data Transfer*

Aarts (2005) described the Scenarios Navigator as a data and information base data-sharing method. The concept in this method can explore a database framework and then use it to solve the convention and reuse of data record from systems (e.g. Expansive database). The situations supervisor available in Scenarios Navigator allows models to have regular access to information and data sources through a generic interface of the models as described in Figure **2.2.**

One of the limitations in this method is that an expansive database will result in a larger structure, which tends to make it difficult to conceal data. Hence, it cannot have the ability to adequately incorporate many system data applications and also embrace the changes to reenactment programming, which can be better handled in the current transformation process.

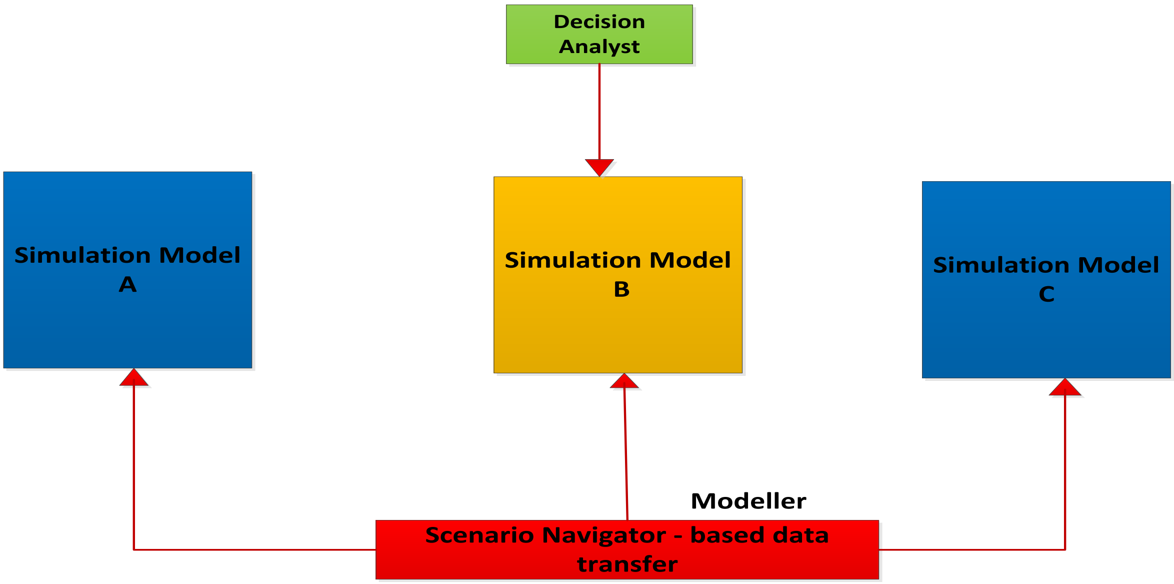


Figure 2.2: Data transfer based on Scenarios Navigator approach

2.6 Modelling and Simulation

Modelling and simulation involves different techniques and technology such as DES. The discrete event simulation packages are types of computer-based modelling software that provide a flexible way to imitate the behavior of complex systems. DESs have been widely adopted in order to analyse, understand, and optimise processes using their structured environments (Kang et al., 2015). Praehofer & Pree (2003): Kang et al., (2015) described discrete event simulation as a model with a ‘class’ of theory and data representation, that is associated with the different modelling element (ME) attributes and internal functions.

Figure 2.3 describes the view of the relationship between models, simulation and system. The difference between modelling, models and the modelling language of the simulation software are mapped and described by the model using encoding modelling language.

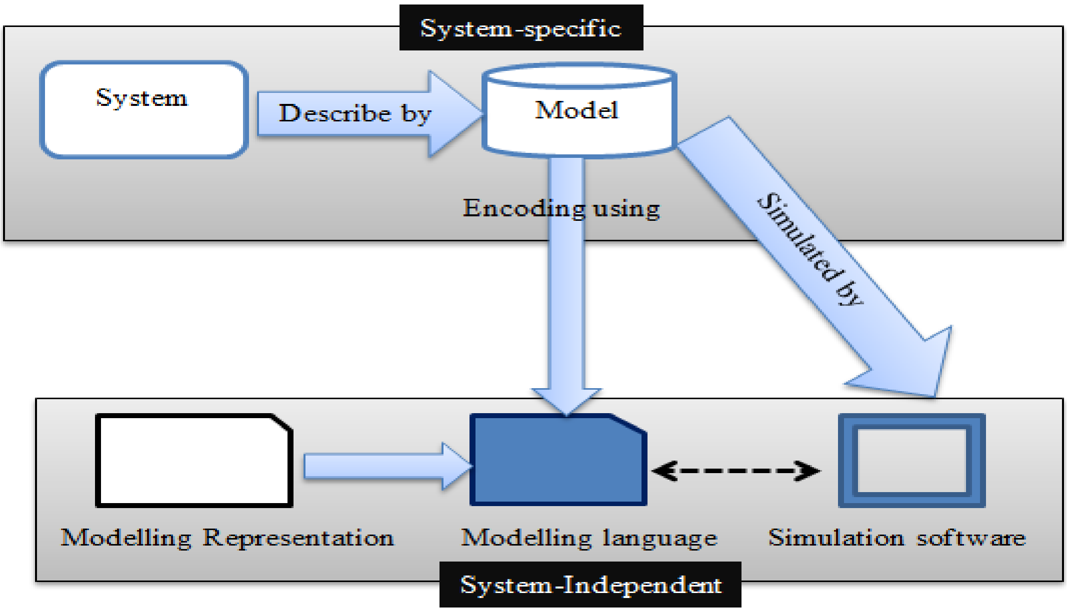


Figure 2.3 Simulation and Modelling concepts

The DES is developed based on the following three aspects (Kang et al., 2015):

1. **Simulation**: Lafortune and Cassandras (2009) describe *simulation* as the method of estimating and evaluating a system model and variable of interest through numerical analysis.

   Ball (2009) also defines simulation as a numerical process used to evaluate the system model, and to analyse associated constraints and the interrelationship of processes. The simulation imitates the system behavior using an apparatus or situation to model information in space and time, thus allowing one to identify the interaction in the model (Ball, 1999).

2. **Modelling:** This is the representation phase of the system in time and space. The model is encoded using modelling language, while the modelling language contains the statements from modelling formalism. However, a model consists of a collection of information views such as resources and activities (Kang et al., 2015). Simulation and Modelling are methods for developing interaction and understanding of a part of the system.

3. **System:** A *system* is a process that performs a range of actions of concern that operate in time and space. In this research, the system refers to system data modelling (Ball, 1999).
4. **Dynamic or Static Data Structure:** Dynamic data are structured to facilitate the structure change at runtime. The assigned elements' values are subjected to change. In a dynamic data structure, more elements can be added or old elements can be removed and replaced with new elements (Garg & Tyagi, 2012).
5. **Deterministic and stochastic:** *Deterministic* refers to a model determined by knowing the relationships among events and states without any variation. This means that the given input data will result in the same output data, e.g. data inputted in the DES will serve as an output to other models. In contrast, *stochastic* is a name used in the simulation field to describe a system or event that is not predictable due to the unexpected influence of random variables. It also refers to system with uncertainty about its value and parameters (Mira & Fernández, 2003).
6. **Discrete and Continuous:** Discrete data only accept certain values; for example: a work entry point in SIMUL8 can be categorised with attributes such as distribution, processing time, etc. In contrast to discrete, the continuous data are not restricted to particular values, but they have the ability to occupy values over a continuous range. They are always numerical and can be an infinite number too. The proposed methodology in this research will adopt discrete event simulation.

2.7 Choosing DES

Discrete event simulation packages are well-established tools in investigating complex systems in manufacturing. The manufacturing industries, in an attempt to keep pace with the reality of global and technology advancement, seek the use of simulation artifacts to model the behavior of their systems that are difficult to understand using manual study. The DES is now accepted as the best option for the design, planning, design and restructuring of manufacturing systems (Lafortune & Cassandras, 2009); the manufacturing industry - in its resolve to meet customer demand with increased complexity in their production systems - has made it their best option.

The usage of the simulation artifact for a manufacturing system is now frequently used to improve the production efficiency and to address various operational problems (Praehofer et al., 2000). As the usage increase, there is increased need for quick and effective deployment of DES models. However, there exist a number of factors that affect the smooth usage of the simulation, factors such as data sharing, lengthy data processing and an efficient platform for data reuse among the simulation tools. A serious factor is the data sharing, which contributes to the lengthy process and more resources investment when data cannot be reused and recycled and be in the right format ready to be used among the different simulation tools. Trybula, (2004), submitted that almost 100% of the commercial simulation tools cannot share their data with other tools and this has not changed, and furthermore there is no effort to change this.

As more simulation tools continue to make their way to market, so also many applications continue to adopt them to simulate their complex system, therefore making the simulation tool modellers build more complex models that also require a large number of data. As a result, this research looks at the family of DES tools and the manufacturing system that was used to demonstrate the use of DES tools.

2.8 Chapter Summary

The problem of data integration and data sharing among the DES tools can be explained in different fields, such as heterogeneous DES systems, incompatible terminology and a lack of governing standard approaches. The future trends show no sign of anything that will improve the data integration for DES tools model data that are heterogeneous in nature.

Due to increasingly complex manufacturing systems and the fact that numbers of DES tools will continue to make their way to market, the number of DES tools will therefore not decrease. Furthermore, the producers of these tools will continue to develop them based on what suits their needs without considering the end users. This will therefore mean data integration remains heterogeneous and will lead to data sharing problems.

As such, homogenous DES tools can be seen as a process that cannot be forced to allow their data to be shared; therefore, research should focus on developing integrated approaches to allow seamless data sharing among the DES.

The current approaches to data integration insufficiently consider how they can be introduced to Commercial DES tools in a manufacturing system; most of these approaches cannot accommodate DES tools and the existing approaches involve organisational, financial and technical risks. Thus, methods that will incorporate different DES for data integration are required.

Chapter 3 - Discrete-Event Simulation (DES) Tool Data Integration

3.1 Introduction

The literature review in Chapter two identified the major problems of DES tools, such as heterogeneous problems and incompatible terminology. That DES tool model data are represented and presented independently of other tools. Hence, allowing the DES different tools to share their data will require certain strategies which include accessing model data integration from a model and a data point of view.

This Chapter demonstrates the DES tools and their models involved in data integration by referring to DES models of a Phone part production line in a Manufacturing System. Current strategies such as Top-down and Bottom-up are assessed and their potential compared in solving data transformation and integration problems among the different DES tools.

3.2 Discrete-Event Simulation Packages (DES)

The DES tools are developed and produced by different vendors that support the simulation. According to Swain, (2003), there are over 50 available DES packages, with many continuing making their way to market, thus making any search quickly out of date. The important aspect of having selected this area is the aspect of different representations for the naming of data by this model, therefore making it difficult for the users to have a more data-friendly DES. However, using the simulation process is always a complex process with the discrete event simulation packages (e.g. SIMUL8, WITNESS, ProMODEL, ARENA, etc.) used to describe the models of the system (Pidd, 2002).

It is therefore difficult to analyse the construct of the DES package language as software continues to evolve. It is necessary to identify how the process interacts and how DES feature representation will help to categorise the packages and their data files so that data can be easily shared between them.

3.2.1 Discrete Event Simulation Usage and Advantages

DES is one of the most widely accepted simulation techniques used in manufacturing and operational research. The simulation models the process of discrete events. This means that entities are passed through between different states with respect to time (Babulak and Ming, 2012). The DES comprises a flow chart that includes a library of building blocks that are pre-defined internally to model processes with a range of distributions (statistics).

Some of the advantages of the DES include improved production or financial forecasting and enhanced short delivery of products and services (Ball, 1999). Furthermore, the DES enhances the reduction in bottleneck with better inventory levels and personnel; the use of DES improves the operation process, thereby increasing overall profitability.

3.2.3 Problems and Opportunities

The rapid change in the business environment and technology development has resulted in different industries and processes (e.g. Manufacturing, business intelligence, education, etc.) adopting DES.

1. ***Manufacturing Production system:***

Advancement in modern technology has made manufacturing systems such as aerospace, electronics, and semiconductor industry process to be more complex due to numerous steps in manufacturing, such as multiple processes of products, parts, complex equipment used multiple sub- assembly plant, downtime and maintenance etc.

Resolving this complexity coupled with the high setting-up cost necessitates the use of the DES modeling tool rather than relying on traditional judgment for the decision making and performance evaluation method (Chance & Robinson, 1999). With customers demanding better and cheaper products in a short delivery time, this makes the production system more complex. The implication is how to subject the process to an automated process and sustain the in-house operation and expertise in the long term to develop their products. Meeting the demand of the customers for on-time deliveries of products is becoming a challenge for manufacturing companies. One available solution for this type of problem is the use of a simulation manufacturing environment for their overall production process, therefore making DES the ideal platform for making this possible (Babulak & Ming, 2012).

Problems

1. **Data Sharing**: The real challenges facing the manufacturing companies is the data sharing between the tools they use in the desire to improve their production quality, reduce operating costs, and shorten development cycles. Reusing existing data rather than generating new data can ensure a huge positive impact on the development cycle and costs (Bishr, 2008).
2. **Heterogeneous Data Sources:** Data sources carry different data models with similar representation of semantics that might be dissimilar with each data source. Moreover, data contain conflicting information (Macura, 2014).

3. **Syntax and Semantics:** Semantic issues are major problems with data sharing faced by many companies using the DES; semantic problems arise as a result of differences in meaning. For example, data from a DES might be carrying an element or a name different from another DES (SIMUL8: Work Entry Point); this is not the way that ARENA will understand it (Amit et al., 2005). In contrast, the syntax is the grammatical structure of the model.

4. **Incompatible Terminology:** For any system to interchange its data, ME and their attributes are the key issues that need to be resolved due to terminology differences. The reason that each model uses different terminology is that it suits their usage and concepts, leading to customisation of models (Carlos et al., 2013). Various misunderstandings and interpretations of terminology for data sharing have arisen due to problems related to the individual application terminology (Lafortune & Cassandras, 2009).
5. **Independence of Data Sources:** Data sources produce for the same purpose but from different vendors and not produced for data integration system. The producer cannot be forced to change their intention toward allowing their tool to share their data with tools from other system**.** As a result, they can ultimately change their functionality without notification to the users (Tatbul & Convey, 2001).

3.3 Discrete Event Simulation DES Packages

The simulation software's producer has produced this tool to suit their needs with different languages and data presentations to represent their application based on their requirements, which tends to define how their process interaction differs from others.

The simulation process is supported by various tools (e.g. SIMUL8, ARENA, etc.), with the market for simulation continuing to grow and new packages making their way to the market, and thereby making the existing method of data exchange to be more complex and out of date (Swain, 2014). This section illustrates the discrepancies or similarities for DES packages using SIMUL8 and ARENA as an example.

1. ***ARENA software packages:***

The ARENA simulation software of Kelton et al. (2007) is a type of commercial discrete event simulation package built on the simulation language and animation system (SIMAN).

The software model of the ARENA process is built by continuing to choose and to place the icon in different types of selected boxes onto a drawing board in the model environment.
The ARENA software can interact with Visio, Visual Basic Application (VBA), and Data Access interfaces (Swets & Bapat, 2000). The Access file interface is used to collect the required data and information described from ARENA.

2. ***SIMUL8 software package:***

SIMUL8 is another discrete-event simulation package used for planning and designing of real production, manufacturing, and optimisation services and logistics.

SIMUL8 simulates a system by using the modified multiplicative-congruential to process and present the discrete entities in relation to discrete time. It also uses XML files and Visual logic (Hauge & Paige, 2002). The SIMUL8 and ARENA tools use different process representations in their models.

3.3.1 Discrete Event Model Process Illustration

Boundary conditions

Current research adopts the DES model that is comprised of all sequences of events contained in the model as presented by the modelling elements and their associate attributes described in Figure 3.1. In this research, the modelling elements are considered as they represent the overall DES tool model and this cannot be changed at the tool level but allows for the definition of external processes such as the manufacturing production line to be modeled at model level.

Model size (modelling elements) and scenarios

The modelling element and associated attributes are presented using two different Model scenarios in the ARENA and SIMUL8 models. The process representations usually presented in a textual data file, which help this research to have the necessary data for the transformation and integration (Narain, 2001). The SIMUL8 and ARENA basic simulation elements are as follows, see Figure 3.1:

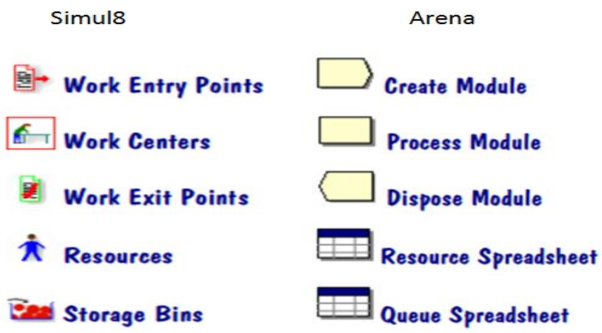


Figure 3.1: ARENA and SIMUL8 basic simulation Elements

1. Create (ARENA) and Work entry point (SIMUL8)

<u>*Create Module:*</u> Create module is the arrival of a job/work item to the model, such as a process that arrives as defined by the user and how often it wants to process the application.

<u>*Work Entry Point*:</u> Work or process application item in SIMUL8 arrives as a work entry point in the model; it allows the user/simulation model to control how many times it wants to create the application or entities and the number required, as seen in Figure 3.2.

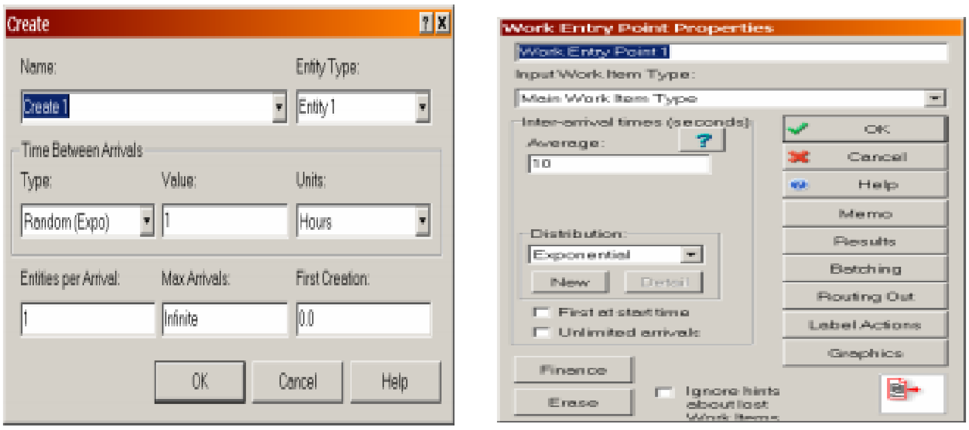


Figure 3.2 Create in ARENA and Work Entry Point in SIMUL8 (Jaret etal, 2004)

Table 3.1 shows the work entry point and creates information as contained in SIMUL8 and ARENA simulation software.

Table 3.1 Work Entry Point and Create Module (Jaret etal, 2004)

Create	Work entry point
Name: Create application name	Name: Enter application name
Time between arrivals: • Value: the value appears when exponential is chosen for distribution • Type- The type in ARENA specifies the distribution probability to represent the value and inter-arrival time. Units- it uses unit to define the parameters.	Inter-arrival times: • Distribution: the probability value represented by the inter-arrival time it can take and defined in the global setting • Average: as relates to the ARENA; is a distribution that also appears when the exponential is selected for distribution.
Max arrivals: Setting limits for the number of entities arriving through the module.	Unlimited arrivals: The entities arrive when the upstream signal is ready for processing.

2. Work Centers (Simul8) and Process Modules(ARENA)

In SIMUL8, work is performed in the Work Centre. The user is allowed to add value, specifying the process, the work it needs to work on, and the time it will work while the distribution can be chosen based on the user's requirement. The user will also need to specify the resources or available resources it requires to work on. In the Work Centre, the user can specify if there is any breakdown and the time it will take for the repair (Jaret et al., 2004); see Figure 3.3.

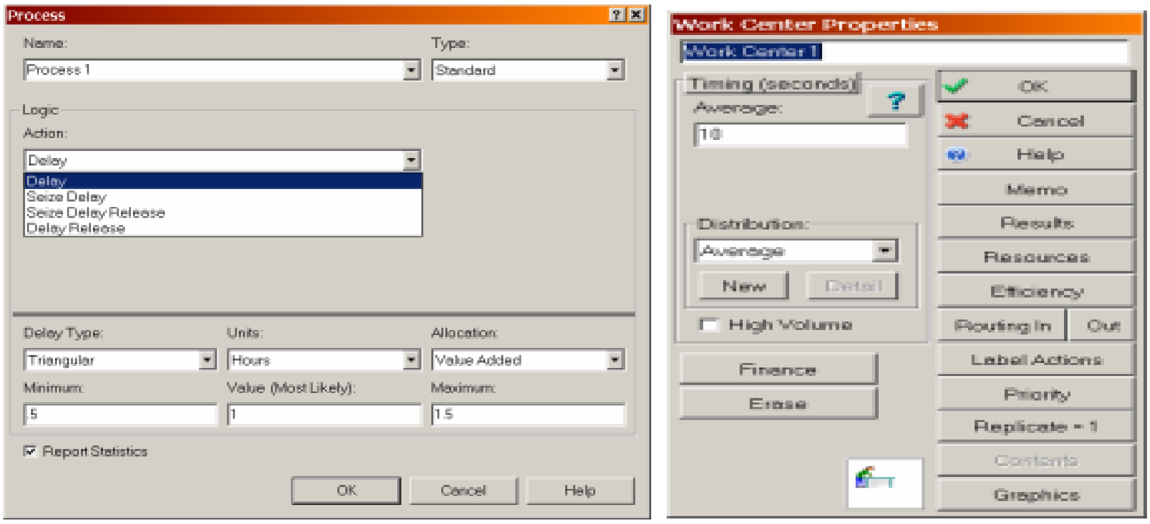


Figure 3.3: Work centres and Process Module (Jaret etal, 2004)

The ARENA Work Centre is modelled by the process module in the model. The user can choose to delay the time requirement, and the user can also specify the time each entity needs to arrive or work on. As related to the SIMUL8 Work Centre that can specify the type and quantity of resources it wanted to use, the delay type has options to choose from distributions (Table 3.2).

Table 3.2 Work Centre/ Process Module (Kelton et al., 2007): (Kang et al., 2015)

Process Modules	**Work Centres**
Name- process	Name- Working Center
Delay: This is the distribution in ARENA and enables the user to choose which work to perform and also to allocate the time and distribution it takes to do the job. • Delay type: *The user can choose the probability distribution that represents the value the work time and duration can take and the value of each are specified (Kang, et al., 2015).* • Units: the general timing units for every process that can be specified in the model. • Allocation: *We use this to specify the distribution probability which represents the values the work and the work time can take and the likely values of each (Kang et al., 2015).*	Timing- The time taken for the SIMUL8 model to work on entity as specified by the model user. • Average: *This will appear once the exponential is selected for the distribution for the process parameter to choose.* • Distribution: *Specifies the probability distribution representing the value the work items can take and the value of each; this changes with distribution, for example,''uniform''changes to the lower or upper bound* (Kang et al., 2015). • Timing value based on distribution: The triangular option is chosen for the distribution.

3. Dispose module (ARENA) and work exist point (SIMUL8)

The modelling process is concluded by disposing it through the dispose module; thereafter, the statistics and total time spent for the overall process can be calculated (Jaret et al., 2004); see Figure 3.4.

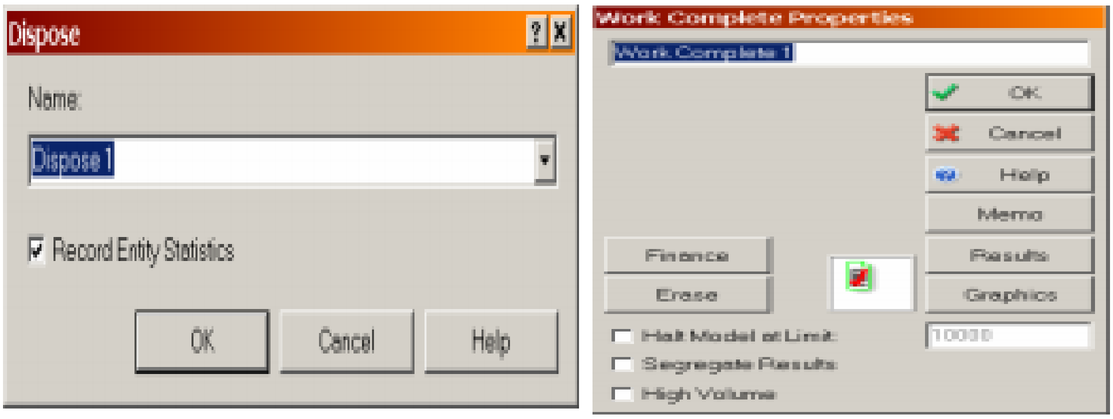


Figure 3.4: Dispose /Work Exit Points (Jaret et al, 2004)

The work exit point involves completing the process in the model where the items or entities leave the system after modelling. Like ARENA, the user can collect the data statistics and know the total time and other performance measures modelled in the system (Hauge & Paige, 2002).

4. Resources in SIMUL8 and resources in ARENA

Both SIMUL8 and ARENA use their resources in the same ways; the user is allowed to define what type and quantities of resources it requires to perform a process cycle.

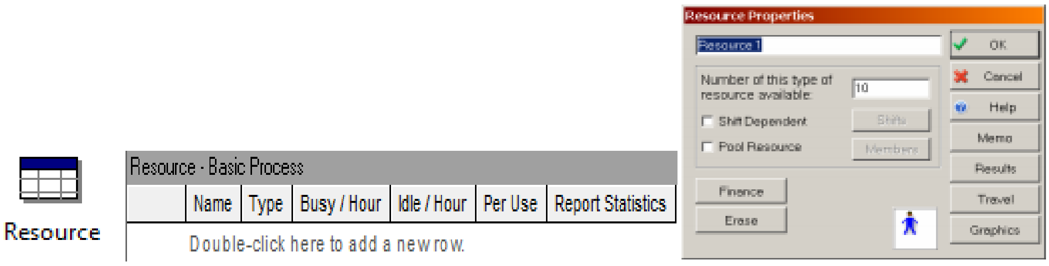


Figure 3.5: Resources in ARENA and Resources in SIMUL8

The resources available can be known, and the resources spreadsheet can tell the user if there is a change in the amount of the resources over time; see Figure 3.3.

Table 3.3 Resources in ARENA / Resources in SIMUL8 (Jaret et al., 2004)

Resources (ARENA)	Resources (SIMUL8)
Name: Resource name can be specified	Name: Resource name can be defined in SIMUL8
Type: The resources need to be specified if it involves fix or shift (schedule) resources.	Shift-Dependent: There is need to specify whether the number of available resources depend on shift (schedule). In SIMUL8, the number of resources available and their duration need to be specified in the shift schedule.
Capacity: Number of entities available and the capacity the process can handle at a particular time.	Pool resource: Specify which part is the resource (set of resources or resource pool).

5. System-Level Attributes

Table 3.4 Simulation Parameters

Simulation Parameters	Value
Simulation time	The simulation run time is defined according to the simulation time desired by the users, and the simulation will stop and terminate when the run time ends (Hauge & Paige, 2002).
Warm-up period	The warm-up period is the specified result collection period and re-runs the simulation with a different random value.

Travel time	This travel time is determined based on the requested numbers of process or orders that are specified in the simulation model.
Shift pattern	Number of teams or time needed to finish a particular job before another process can take over.

3.4 Relevance of DES Simulation Data Transformation and Integration

The relevance of data integration can be shown with an example in Figure 3.1, from two discrete event simulation packages (ARENA and SIMUL8). The DES packages adopted different data representations of the same or similar tools with the use of their domain definition and representation that suits their specific needs, therefore making the integration of data between them a complex task.

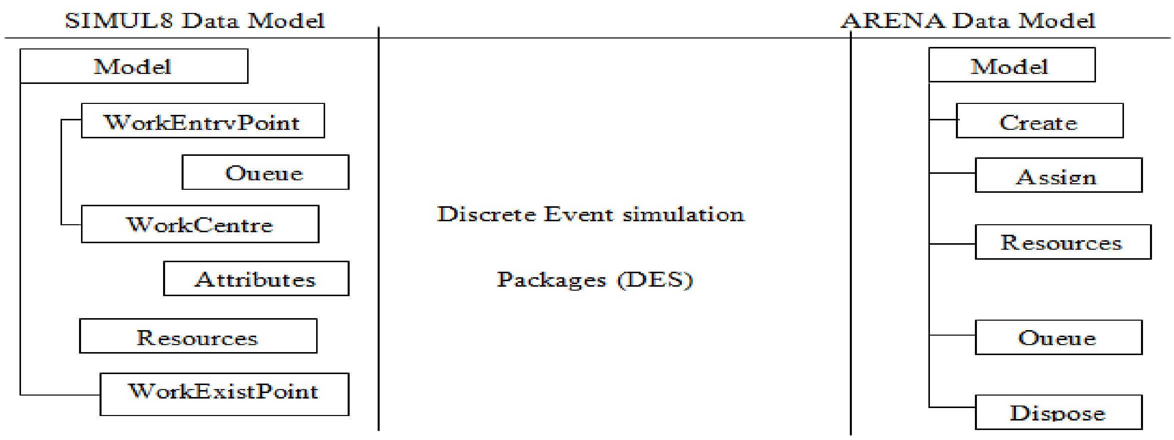


Figure 3.6 Different Data Models for DES

As seen in Figure 3.6, the data model of DES, the SIMUL8 and ARENA are similar but not equal due to each tool specification and requirement. The DES data model contains complex data types of all elements with special features: Work Entry Point, Work Centre, Resources and Work Exit Point etc. In contrast to this, the ARENA data model contains: Create, Assign and Dispose etc.

3.4.1 Definition

Defining *data integration* in the context of this research involves the combining and sharing of data from different DES sources, which are stored using various files (e.g. XML and Access files, etc.), and preparing it to be ready to be consumed by other system models (Bishr, 2008).

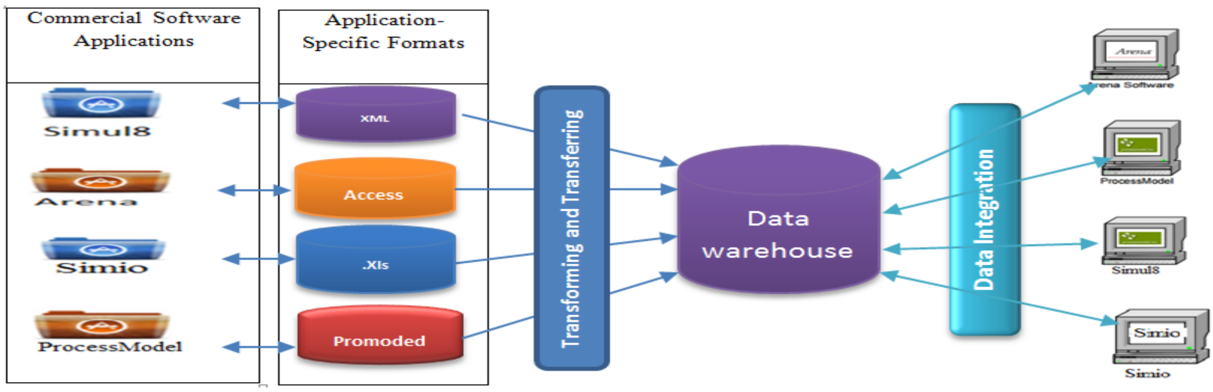


Figure 3.7 Data Integration Design

The importance of this integration is merging two companies' processes to provide a good view of the company datasets.

3.4.2 Classification of Integration

For the current data integration, this research accesses data application integration at the discrete-event simulation tool modelling level, with the requirement for a particular data model. The data are only known by the specific domain because their representation, functionality, and expression differ from one tool to another. APPENDIX 1 is used to illustrate the difference between data models underlying the two DES applied, for example during SIMUL8 and ARENA software development. This research focuses on enhancing data sharing between discrete event simulation packages. Available approaches involve established data integration as either Top-down or Bottom-up.

3.5 Existing Data Integration Approach

In this section, existing strategies for data integration, their advantages and limitations will be argued; also the chosen method for this research will be discussed.

3.5.1 Bottom-up Approach

The bottom-up strategy explains the unplanned data exchange strategy and can be adopted by any simulation tools used in industry as a starting point in the data sharing process. The concept of the bottom-up strategy is done by adding all data in the tool model and creates an integration from one tool to another (Bergert, 2007). Although almost all of the DES tools provide some interface mechanisms to import and export their data through the use of a text-file base format, the bottom-up strategy does not consider the general concept of various tools.

3.5.2 Top-down Approach

The top-down strategy consists of establishing an independent generic data model of the DES tool in order to be accessible for the integration process. The top-down strategy is more focused on data consistency stored in order to remove redundancy of the same elements of the data in the model. However, the generalisation of the concept does not mean that all information required will be captured; therefore, the generalisation needs to be extended to enable the data sharing, but the companies that use these tools have minimal control because the tool has been customised by the process. Drath (2010) explained that automation modelling language (ML) is a good example of a top-down data integration strategy, but does not take care of exchanging and interpreting companies' specific requirements and standards. In contrast, the users of these tools are reliant on the data sharing being resolved by the company.

3.5.3 Limitation of the Top-down, Bottom-up and MDDI

Taking the two examples at this stage, the current data exchange is compared and contrasted in Table 3.5.

Table 3.5 Advantages and limitations of some of the existing strategies

Top-down	Bottom-up	Model-Driven Data Integration (MDDI)
Advantages: It allows data consistencies	**Advantages**: Accepts tools and their data requirement and allows different tools to define data to be utilised.	**Advantages**: - Proactively incorporates various models and utilises metadata across the data integration process. -Provides an integrated standard development method for a good transformation process. - Allows for the definition of relationships between different models at a conceptual level. - Provides an integrated development method for support, interoperability, integration, portability, reusability and adaptability (Hyeonsook et al., 2009).
Disadvantages: Does not allow data to be incorporated at a later date after development.	**Disadvantages**: Changes are difficult to be implemented in the data model.	**Disadvantages**: Requires an external template before a new data model can be introduced, this might compromise the integrity of the data sharing process because there might be more or less data shared unnecessarily.

The advantage of the top-down approach is always ensuring that all data are consistent, but data-specific definitions and requirements of individual tools' data resources are not considered at the initial phase of the top-down stage, as data cannot be incorporated at a later date. Therefore, issues might arise in meeting the data sharing of individual model data, which sometimes can result in a complete redevelopment; this will be considered in the current research.

The bottom-up strategy allows for the understanding of each discipline by steps and allows the integration of data from one tool to another. However, rapid data sharing increases complexity, therefore making it difficult to achieve automated data exchange.

Additionally, a new technique is required that can combine the advantages of both bottom-up and top-down techniques. Therefore, the technique in this research is the Model Driven Data Integration (MDDI) method that will introduce the new technique that is applied to the DES tools.

3.6 Selecting the current approach for Data Transformation and Integration

Model-Driven Data Integration (MDDI) is an Object Management Group Specification (OMG) used in a model language transformation and has been adopted for transforming data from one source to another; for example, model-driven (e.g. Model-driven method, model to text, model-to-model, etc.) data transformation (Haas, 2003).

According to reviews, model transformation is a unique way of mapping and transforming the heterogeneous data sources into a unified view (Jianbo, et al., 2015).

The Model-Driven Data Integration for discrete-event simulation packages is different from bottom-up or top-down strategies, because it utilises and incorporates metadata into its data integration process, therefore reducing the complexity identified in top-down and bottom-up techniques while also providing the standard integration method. Therefore, this research introduces the MDDI framework for discrete event simulation packages' data sources. As a result, two example scenarios will be presented in this research for the transformation and integration: defining the concepts of DES models in a UML class diagram, mapping of model to models, with an input source model and the target output models in confirmation with both models' metamodels, executed in an eclipse model transformation environment.

3.7 The Model Driven Architecture

Figure 3.2 illustrates the overall MDDI software architecture (the rectangular shape represents the software components, and the arrow shows the flows of data and metadata). In the context of this research, the source and target models contain the relevant data for all the DES sources as seen in Figure 3.8; the two data models represent the models. The MTL transforms the information in the source model to the target model that is based on a metamodel with other processes described in the steps below.

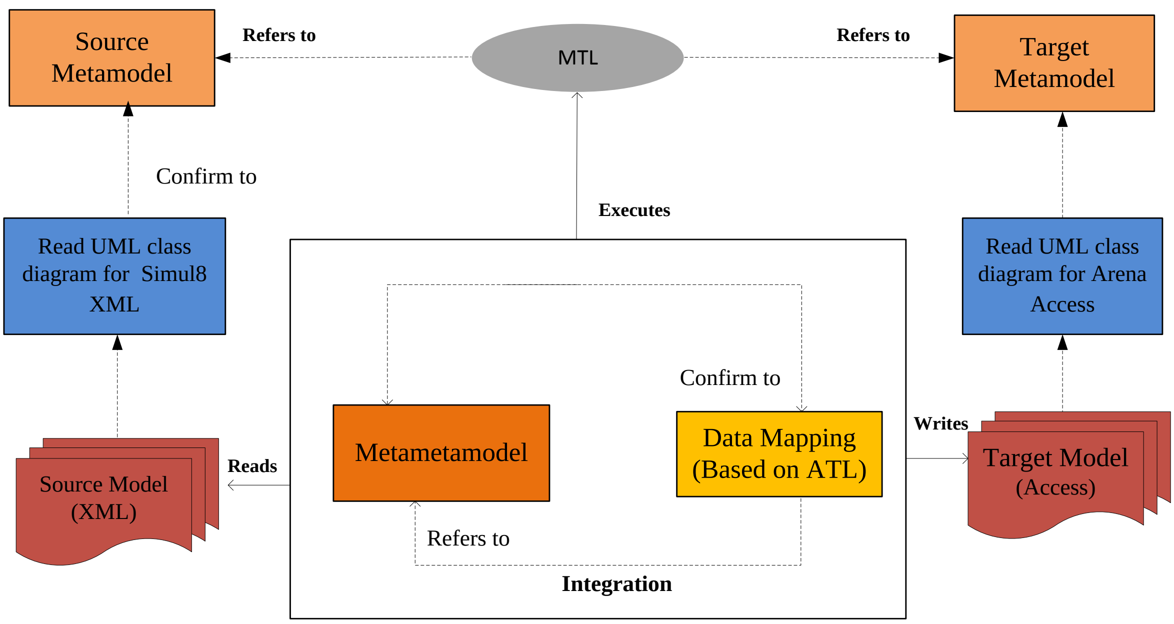


Figure 3.8: The Model-Driven Architecture

3.8 Simulation Meta-Modelling

A metamodel is the definition of the model while metamodelling is the procedure of generating the metamodel. The metamodeling is the construct, analysis and development of rules, frame, and models applicable for modelling a predefined problem class, and is therefore useful in this research to describe the model of discrete-event simulation models. Metamodelling is an important process to describe the DES modelling language and the related interaction process. The entities and their relationship models, as well as event-driven chains, are all defined in a metamodel, indicating the model of the model in the simulation software (Kirchner & Jung, 2007). Identifying the relevant process in data transformation and integration is the first unique way to achieve a flexible technique; the metamodelling provides the support to understand the model.

Several researchers have adopted metamodelling for modelling different models and the goal is to primarily define the semantics and the concepts of different systems, as well as specifying their classes and their subclasses to support the integration process using Unified Modeling Language (UML) to present the DES model. This method is thereby adopted in this research (Silver & Miller, 2006).

3.9 Chapter Summary

Techniques such as bottom-up and top-down were investigated, but the Model-Driven Data Integration (MDDI) approach has been chosen for this research. Researchers have on different occasions adopted different techniques for transforming data to allow the sharing among systems. Finally, from the literature, the transformation and integration of the heterogeneous DES data sources using the MDDI approach is the main focus of this research.

Chapter 4 - Model-Driven Transformation Method

4.1 Introduction

This chapter presents a new approach to data integration in the DES tools model based on the new Model-Driven Data Integration Approach. It enables consistent a data model to be established which fulfils the requirements of the DES tools involved in the manufacturing production line process.

A method for determining the model data and details of the manufacturing production line data as the major elements of the MDDI data integration process is illustrated. At first, this is undertaken specifically for data occurring during the DES model design of the process of manufacturing phone part production lines. After this, insights and experience gained from the DES tools are then used to formalise a general description of the MDDI strategy.

This research investigated how this can be achieved through the following:

1. **DES data collection**: Development of DES model use to investigate different process and data concept to:
   i. Generate the model data through the elements and attribute as presented by the DES tools involved, and also generate their interrelationship through the simulation process.
   ii. Apply manufacturing production line data to build the DES model based on the data model of the DES.
   iii. Use the Eclipse transformation and integration environment using the MDDI method to achieve the transformation process.
   iv. Collect the model data through the manufacturing production line.

2. **Data Integration**: Uses the model driven for data sharing among the different DES against other methods such as top-down and bottom-up for flexible method.

This chapter begins with the research methodology and the framework adopted to achieve the set objective in Section 1.6. Therefore, the research problem is identified in terms of the elements and attributes used in building the evaluative data model. The chapter shows details of the tool used in the transformation and integration process as identified in Chapter 3. Furthermore, Chapter 4 also presents the data collection details, and the steps taken to achieve the overall research method. Finally, the chapter presents how the manufacturing production line data is used to validate the model.

4.2 Tool

Eclipse modelling framework is an open source ATL (Atlas transformation language) tool developed by the OMG group that provides an Ecore Modelling Concept. The models are now a major part of engineering (e.g. Software Engineering), the model-driven engineering considered models as the first class data entity and also consider different ways in handling tool models. The choice of this tool is very useful because it provides the developers with sets of operation to enhance the model manipulation and analysis towards achieving data integration. In this research concept, the model driven transformation is the central operation for the modelling of handling data; it also makes it possible to specify a number of target models based on the source model data (see Figure 4.1).

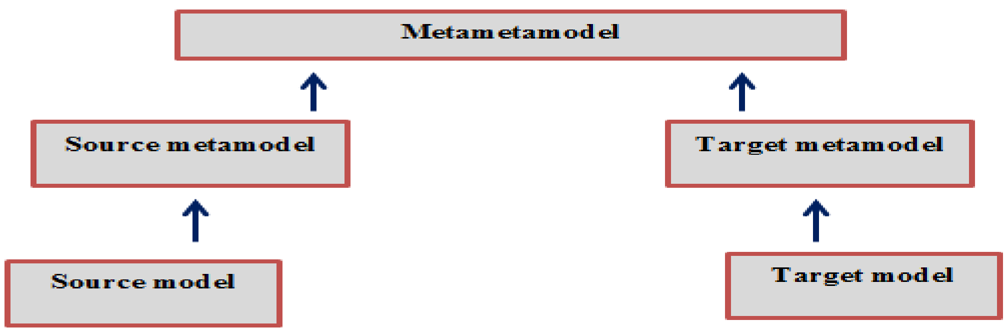


Figure 4.1 The General Picture of the Transformation

The main advantage of the Eclipse modelling framework in model driven data integration is as follows:

1. **Model transformation:** An experiment using model transformation will provide a means for specifying the way to target models for a source model (Douglas, 2006).
2. **Design metamodel:** An Experiment using a metamodel allows the definition of the model semantics of the tools in conformity with the transformation metalmodel of the model, as described in Figure 4.2 (Douglas, 2006).
3. **Conceptual model:** It also provides the overall view of the conceptual model transformation and matched with the ATL language, which in this case allows for the total accommodation of all concepts defined in the model that are involved in the transformation.

Disadvantages

(a) Accessing DES data model requires more time

(b) Data collection can be extensive

The advantage of using the ATL tool outweighs its disadvantages from the viewpoint of this research consideration, therefore, ATL model driven data integration environment was used in this research to model the DES simulation model data within a manufacturing production system data for both data model and manufacturing system scenarios.

4.3 Proposed Research Steps

The first step for this research is to develop a method for transforming the data originating from different DES, so that it can be consumed back by another DES, therefore enabling data sharing among them. The following steps to achieve the aim of this research are identified in Figure 4.2.

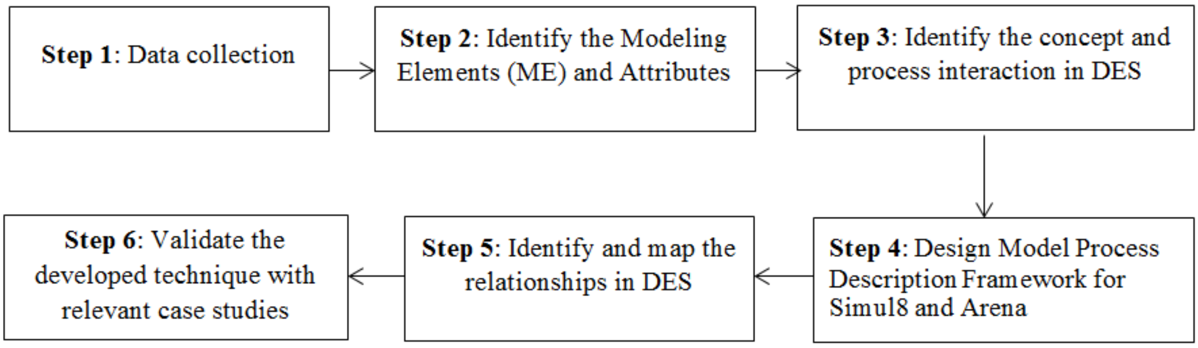


Figure 4.2 Research Method Steps

4.3.1 Step 1: Data Collection

The data needs to be collected for the development of discrete-event simulation models in a real manufacturing environment, meaning the development of two different models representing two simple manufacturing processes. The DES represents ‘‘**Five Modelling Elements**’’ as described in Section **3.3.1,** meaning artificial data were generated in a wide range of concepts defined in the DES models and was used throughout the development process, while the actual manufacturing data was used in the validation stage..

The present data represent the function and description of the model as described in APPENDIX 1, 2, 3 and 4. All the discrete event simulation tools allow model building that changes in a state at a particular time and at particular events. The DES is composed of a network of queues and other events and activities, for example, manufacturing system comprises of buffers, and entities involve sets of attributes which represent the manufacturing system. All entities passed through the sequence of events and later exit or enter the model. The model activities also compete with the available resources such as the raw material and operator of the production of the machine.

The DES simulation model involves the event route, the event list, the simulation state, a clock and the simulation final execution. The event and state routine change are derived from the simulation model, the end simulation execution is used to simulate the model through identification of the time in model clocks and also in choosing time that represents the event and the next event. For example, change can occur in the simulation state such as termination of machine production in the production activities or using a buffer (queue) to store an entity or to prepare a new event, for example an entity arriving in the model. This can be done in a production life cycle until the terminate end run condition are met. All the DES models are based on the conceptual framework and process interaction which are the developers need to use as a guide for the development, for example manufacturing production of products. The DES possesses a different representation and widely differs in terminology, for example, a DES model might be carrying an item and another carrying entity. In this research the DES model is carrying elements activities, queues, resources, entity and process exit point, and the model behavior is considered as a set of rules governing the activities of the queues which are different among the DES, example, rules such as when an item leave a machine to enter buffer.

Therefore, the current method is applied to the manufacturing phone part production line. The data structure from different DES are analysed and with interest and regard to their differences and similarities.

4.3.2 Step 2: Identify the Modelling Elements (ME) and Attributes

1. **Modelling elements:** The modelling elements consist of work entry point, work centers, process; create modules, queue, work exit point and disposes, etc.
2. **Attributes:** The attribute consists of inter-arrival, processing time, distributions, capacity, cycle time, batch size, etc.

The details modelling elements and their attributes are collected and analysed to create a relationship among the DES and to identify those elements that proved to be heterogeneous (Babulak & Ming, 2012).

4.3.3 Step 3: Identifying the concept and process interaction in DES

Two DES products (SIMUL8 and ARENA) were adopted as case studies in this research and each was used to model a simple process of manufacturing a product as described in Step 1.

This simple process was applied to both tools as simulation parameters, and entities such as modelling elements, attributes such as queues, workstations, resources, arrival, and exit are analysed from the XML file generated from the SIMUL8 and Access files generated from the ARENA DES in order to have a common understanding of both packages.

Here, this step is intended to determine whether there are any interactions and relationships between the DES models.

4.3.4 Step 4: Design Model Process Description Framework for SIMUL8 and ARENA

The corresponding concept was evolved into a class description, the process is represented in a UML class diagram to describe the class and properties and also to control the data flows. The metamodel described is used to map the model information to enable the transformation to take place. The UML is a standard object management activity diagram used to present the simulation software design and processes. The UML activities support the semantic activities and therefore, support the flow and control of data behavior.

4.3.5 Step 5: Identify and Map the Relationships in DES

The set of relationship of the model (source) and other models (model) is identified for mapping together to enhance the achievement of the transformation process. The elements defined in the model A (source) are expressed and mapped onto the target model.

For this reason OclModel- **(module** SIMUL8ToARENA Mapping such that **create** OUT: ARENA **from** IN: SIMUL8) is adopted. It expresses that each element of the model (source) must map to an element in the target model.

Once the relationship is mapped, then the model transformation will form the final output of the transformation by defining the condition that can be applied and the result expected. The transformation contains two elements: A set of elements to match for a model 1 (ARENA) that will be an input as a source model of the transformation.

Other sets of elements to be matched for model 2 (SIMUL8) to be used as a target model produced by the transformation. The objective is to present the case in the UML class diagram (metamodel) for SIMUL8 and ARENA models to model transformations written in ATL. Initially there exist data in a text describing a list of modelling elements. Now the objective is to transform this into another text describing a list of modelling elements.

4.3.6 Step 6: Validation of the Developed Technique

In terms of testing the transformation and integration technique, two DES products (SIMUL8 and ARENA) were adopted as case studies to validate a manufacturing production line data. This step is concerned with testing the transformation and its validity to real-world scenarios. The output data from SIMUL8 was transformed within the Eclipse Transformation Environment and consumed by ARENA. Therefore, this transformation method can make the DES to share their data. For instance, a phone part production line with different products was chosen for the validation.

4.4 Experiment

4.4.1 Step 1 Experiment

Discrete event simulation (DES) is developed based on 5 modelling elements presented in the model which represents the concept of the model and parameters as described in APPENDIX 1 and APPENDIX 2 respectively. The data collection process can take a quite significant amount of time within the modelling development, the time taken to collect data is up to 30-

38% of the project life cycle (Romeu & Kolmogorov, 2003), research has also confirmed that DES tools have limited process for data sharing and usually present no support for data management (Skoogh, et al., 2012).

4.4.1.1 Current Data Model Comparison

The manufacturing production line involves usage of a variety of DES tools for the identification of the phone part production line. The essential data models that underline two DES tools are analysed and compared with respect to the structure of their data and their element and concept application. The two DES tools considered are SIMUL8 and ARENA.

The two models were set up on the data models representing a complete phone part production line developed through the two DES in its different structure levels and compared with respect to their differences and similarities.

APPENDIX 1 and APPENDIX 2 provide the useful visualisation of the data structure and their corresponding elements. The corresponding elements in the two DES differ in their representation and nomenclature. For example, some of the distribution in SIMUL8 tool returns an integer and that of the ARENA tool returns string and contains more attributes than another.

Some other differences are names such that the distribution in SIMUL8 is called arrival type in ARENA. However, at the operation and hierarchical level the element can be matched together between the DES tools. This difference is the changing process because each modelling element in the DES differs in naming. Accordingly, this research has dedicated rules that define their relationship, as shown in Figure 5.3 and 5.4 respectively.

In summary, the literature shows that there is a difference between the DES data models as they can be similar but sometimes not equal. Apart from the naming differences of the tools, there are some issues such as mapping the corresponding elements which are hampered by nomenclature that are meant for similar elements in the model. However, the rule was introduced in the Eclipse transformation environment to allow a corresponding element to align with each other, which are the impending data sharing as shown in Chapter 3.

4.4.1.2 DES Data of a Phone Part Production Line

The underlying data model and their relation to the DES tools leads to sets of corresponding elements and their interrelation as seen in APPENDIX 1 and 2 respectively. These elements, as presented for simple manufacturing process as shown in Figure 4.3 and Figure 4.4 and reflect the basic structure of manufacturing phone part production line and the task of production. This research adopted the analysis of the manufacturing production line units and applied it to the modelling element as the basis for data model of the DES. The functional element and their hierarchy of the production line is used as the basis for the data model which is what is shared among different DES tools used for production manufacturing as presented in Figure 4.3 and Figure 4.4.

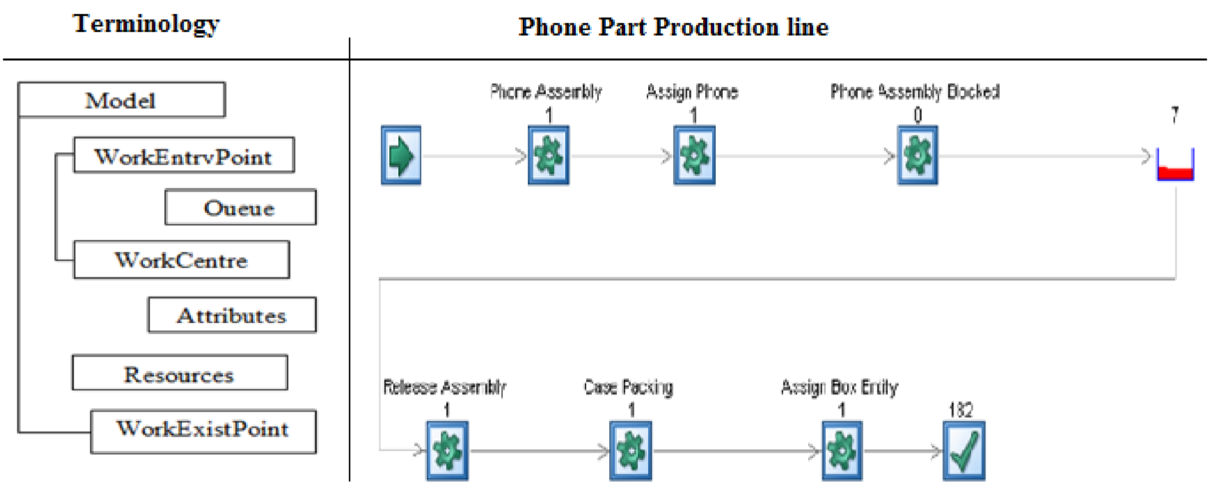


Figure 4.3 Data Models for phone part production line shared by SIMUL8 tool

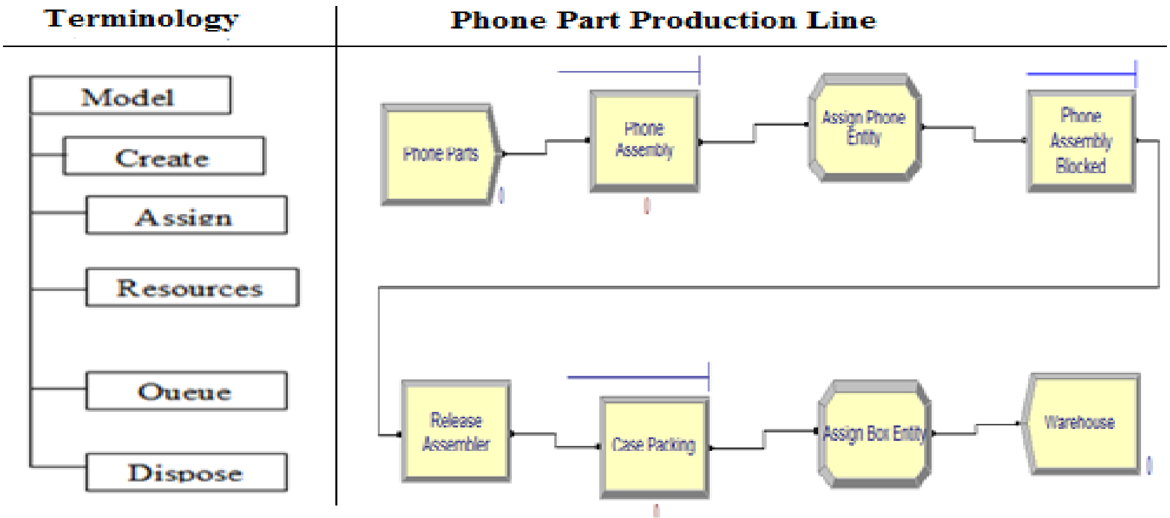


Figure 4.4 Data Model for phone part production line shared by ARENA tool

All DES tools involved in the manufacturing production system share a common understanding. The Figure 4.3 and Figure 4.4 above basic functional model structure and data formed the data of a phone part production line. The information stored in the model data can now be shared among the relevant manufacturing industry because this can be stored in the model data that is underlying the DES tools.

4.4.1.3 Examples of DES Data Models

The DES data models as presented above contained all the data produced and required by the manufacturing processes involved and additional data as contained in the model and relevant to all DES models involved.

In this research, the example illustration about the model data which is ready for exchange between different DES tools in the manufacturing production line using phone part production process line as a sample. In SIMUL8 and ARENA models of the phone part production element, such as modelling element and their associated attributes are described in detail.

The production builds upon the information from the data model from the DES phone part design regarding the process, such as Modelling element (ME) type, ME name, processing time, distribution, resources, travel time, cost, queue capacity, inter arrival time are specified. This also depends on the nature and sequence of the event as describe by the production process and their sequence of operation. An example of this data is shown in APPENDIX 4 and 5.

4.4.2 Step 2 Experiment

Table 4.1 illustrates the ME used in the DES models; the MEtypes are defined by each of the models to represent its internal process. The ME shows the source and the target and gives the direction and the relationship of the data mapping.

Table 4.1 consists of the relational Tables, modelling elements, and their corresponding attributes for SIMUL8 and ARENA model data.

Table 4.1 Modelling Elements (ME) and Attributes

	ARENA	**SIMUL8**	**SIMUL8 source information**	**Mapping**
Model Concepts	Entity (Type) Queue Resource Variable (Entity) Attribute	Inter-arrival times distribution, average Name, Timing(distribution, average) Pool resources Shift dependent Capacity	Type = xml	SIMUL8 Entity Type, Inter-arrival times distribution, Name, Resource, Variable, Timing (distribution, average), (Entity) Attribute

Activities	Assign Create Decide Dispose Process (includes Delay)	Work Entry Points Work Centres Resources Storage bins Resources	**ARENA Source information**	
			Type: access file File name: model file.*.doe)	ARENA Assign, Create, Decide, Dispose Process includes Delay, Work entry points, Work centers Resources, Storage bins Resources.

4.4.3 Step 3 Experiment

Models (e.g. ARENA and SIMUL8) are known to have their individual modelling elements with their associated attributes. Investigation suggests that data are structured differently and in different formats by DES makers. Modelling elements (Figure 4.5) and Attributes (Figure 4.6) summarise the requirements for the models' information.

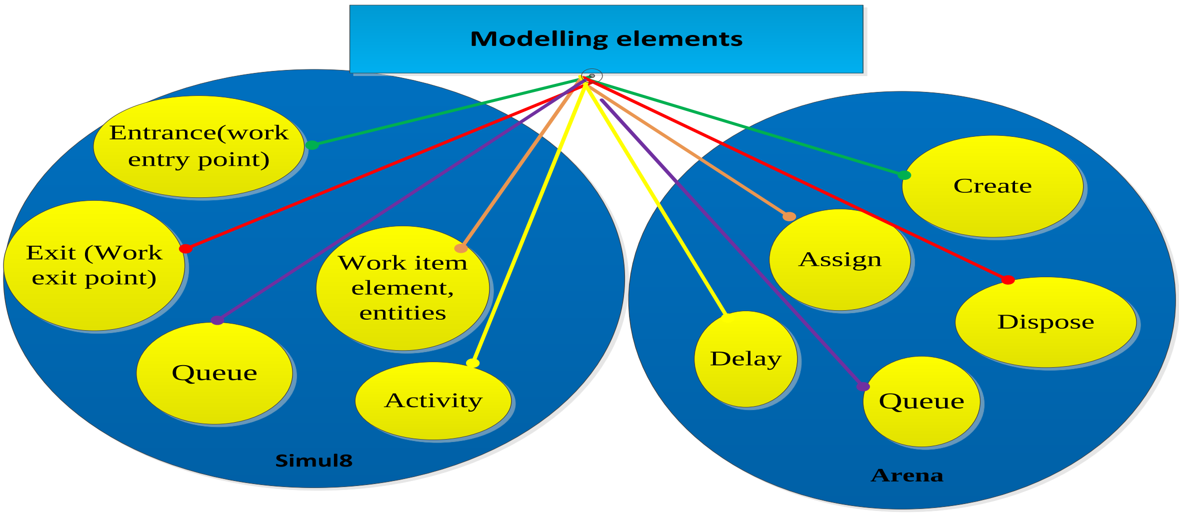


Figure 4.5: Modelling elements' interaction diagram

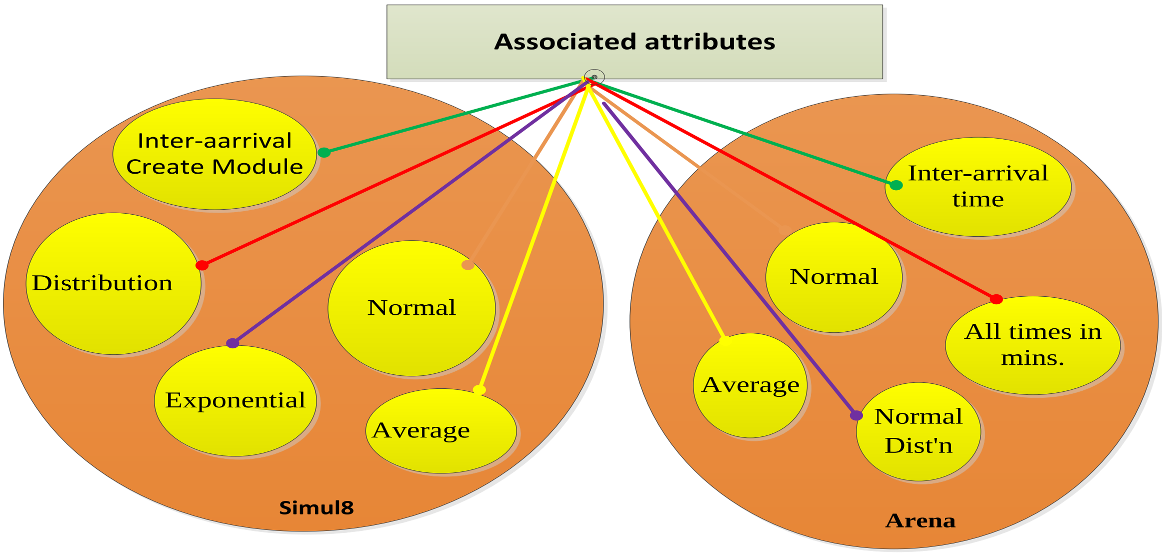


Figure 4.6: Associates attribute interaction diagram

4.4.4 Step 4 Experiment

The metamodel transformation technique uses the UML (Object Management Group (OMG), 2009) to model, interchange and manipulate the metadata of tools. The UML provides the metadata language that is used for the metadata modelling.

Therefore, this research presents the results of MDDI in IMUL8 Process Framework Metamodel (SPFM) and ARENA Process Framework Metamodel (APFM), see Figure 4.7 and 4.8).

4.4.4.1 SIMUL8 Process Framework Metamodel (SPFM) and ARENA Process Framework Metamodel (APFM).

The SPFM and APFM describe the relationship model elements between the two models; the relationship metadata leads to data mapping. The metadata is composed of essential elements, and structures of the models (SIMUL8 and ARENA) process framework model. The transformation and source model are achieved based on the SPFM Metamodel, (Figures 4.7 and 4.8) for SIMUL8 and the ARENA process Metamodel.

The frameworks are achieved by using the Eclipse modelling environment with the definition of classes of element and associated attributes using the concept interactions as presented by the DES.

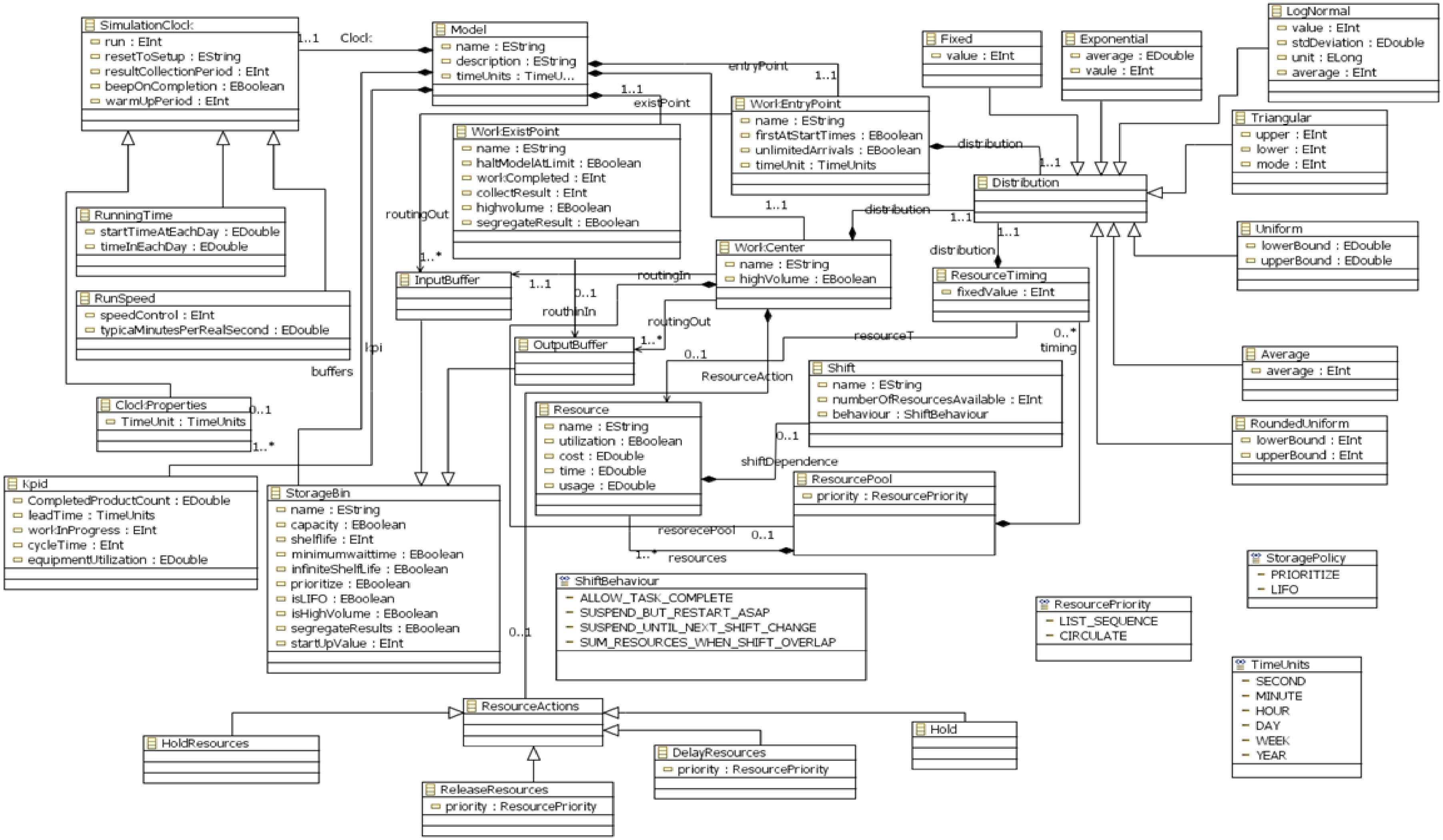


Figure 4.7: SIMUL8 Process Framework Metamodel (SPSM)

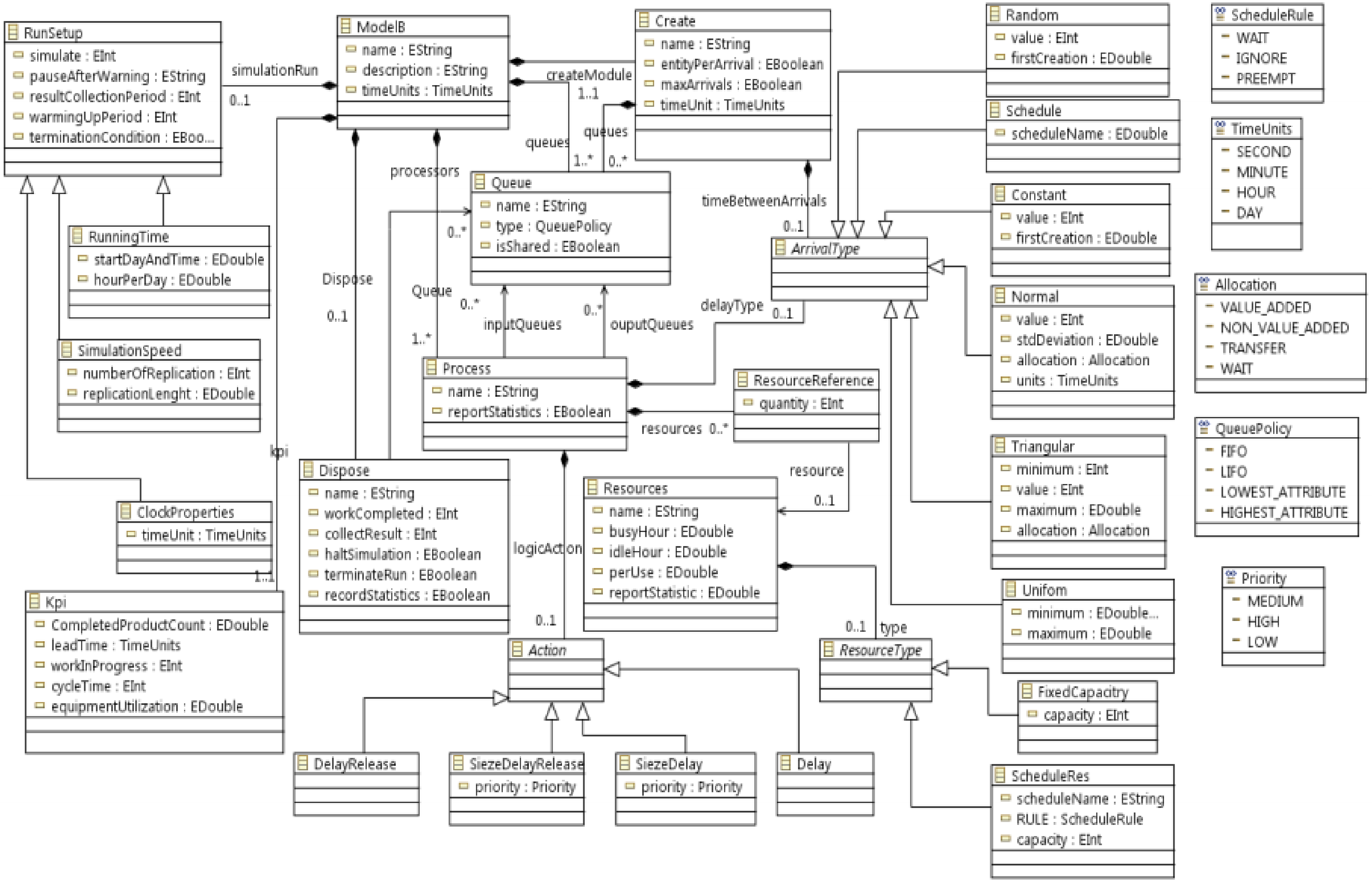


Figure 4.8: ARENA Process Framework Metamodel (APFM)

4.4.5 Step 5 Experiment

Atlas Transformation Language (ATL) has a toolkit and a transformation model language that understands different concepts as defined by various tools. ATL offers a means of producing a set of data (target models) from a set of data (source) (Hyun, et al., 2013).

The ATL would be developed in the Eclipse Platform and environment, which is an integrated and transformation environment of the ATL that offers a number of standard development processes to make the development of ATL transformations easier (Djamel, 2014). The ATL platform provides a number of models (targets) from a particular source model using transformation rules. The rules define how the source model and its elements are navigated and matched for creating elements of the target source models.

4.5 Summary

The Model data are identified by developing the data structure underlying the applied DES in the manufacturing production line process. The development and identification can be done by matching all relations of the elements contained in the data structure, as demonstrated with reference to the manufacturing process of a phone part production line. In this case study, the model data structure matched all the production line structures.

This means that incorporating this model data structure in all applied DES tools allows for the data sharing concerning the DES tools and the application of the MDDI approach to the manufacturing process production lines, thus allowing the sharing of data between different DES.

Chapter 5 - Experiment Results

5.1 Introduction

This chapter presents the achieved results while carrying out the experiment design; the relationships between the different discrete-event simulation packages were identified and used for transformation and integration and consequently used to validate the manufacturing real word data.

5.2 Experiment Scenario

This section presents the characteristics of phone part production lines and provides an introduction to their DES model design process. This includes an illustration of the DES tool model data and information produced throughout the development process. Moreover, the DES tools applied to create these data and information are presented with respect to their degree of integration and the way they are able to share data.

The new concept for the data integration based MDDI has been implemented in the Eclipse transformation and integration environment, as described in Section 4.4. The production lines are typical examples of highly special purpose production system. For the integration of the phone part production lines to work together, the DES tools employed need to work together. To accomplish this, the DES tools employed by the different companies must be able to share data. To enable users to introduce, maintain and enhance the integration solution, the developed MDDI builds upon the well-known environment (Eclipse).

The Eclipse stored all the data in a shared folder on its environment where it can be easily accessed by all the DES tools involved. The Eclipse uses its built-in Windows user environment, in which it works automatically in the background.

5.2.1 Characteristics of Phone Part Production Line

Figure 5.1 shows the layout of a typical phone part production line. It serves as an example to explain the phone application in general which varies in required space depending only on the number of producing units.

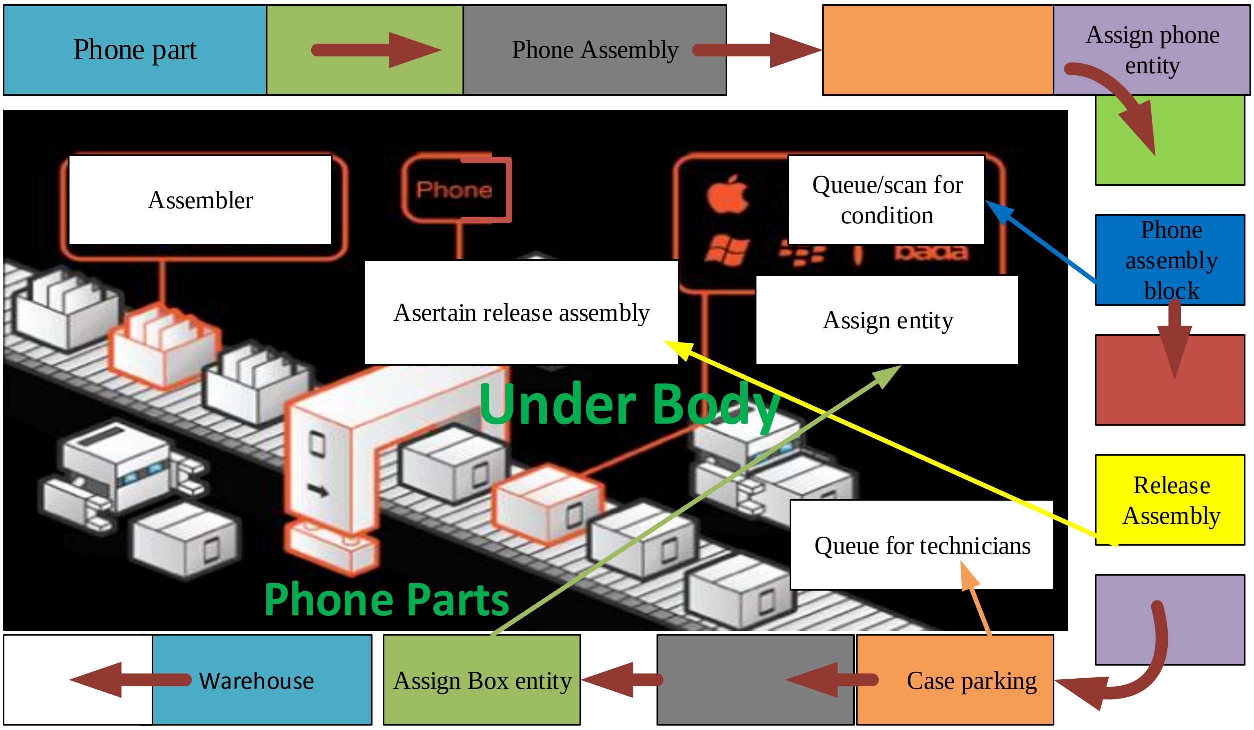


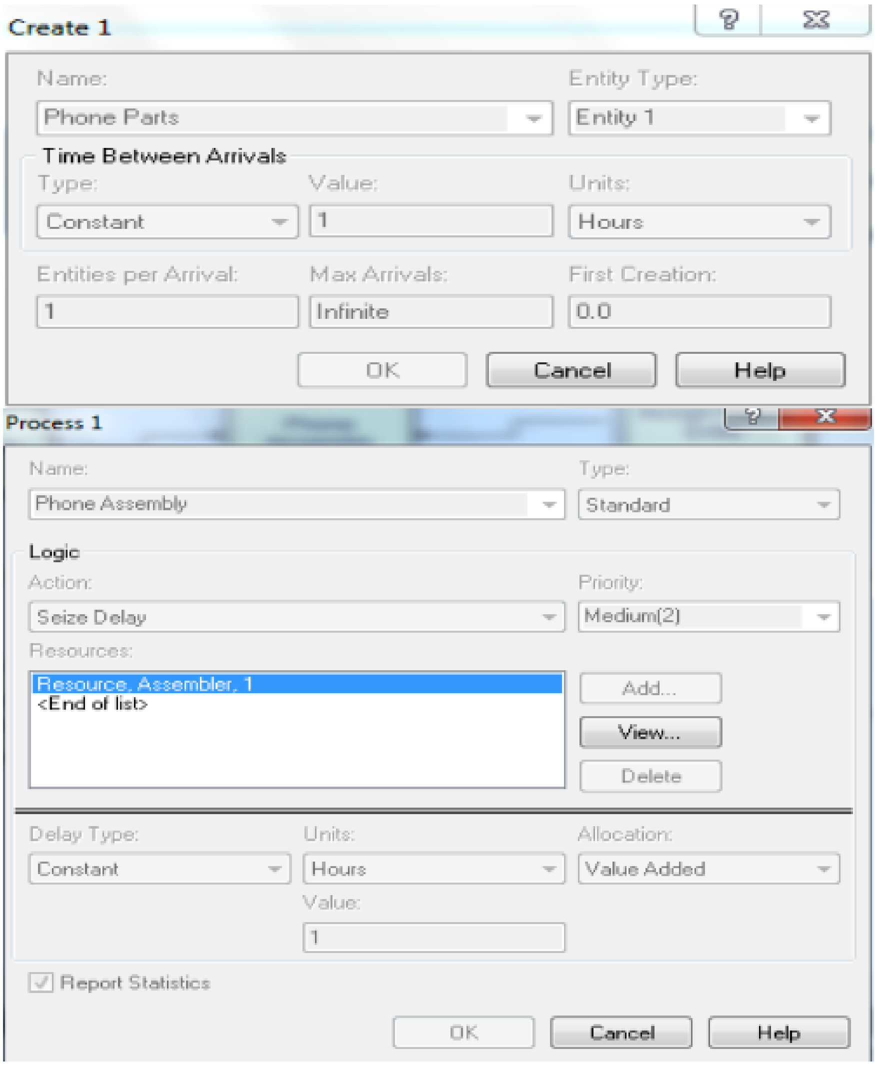
Figure 5.1 Typical Phone Part Layout

The colored rectangles are used to represent seven major assemblies and the white shapes are used to represent the sub-assemblies of a Phone part production line. The production is arranged so that the phone part can be seamlessly transferred to the successor sub-assembly.

Each sub-station is divided into minor sub-assemblies, the underbody sub-assembly contains the phone part assembly system. This assembly is transferred to the shop floor assembly line where they are packaged together. Then the under bogy is transported to the warehouse.

5.3 ARENA to SIMUL8 Experiment

The different DES tools (ARENA and SIMUL8) used by the manufacturing industries which participate in the data integration process of the phone part production line are shown below.



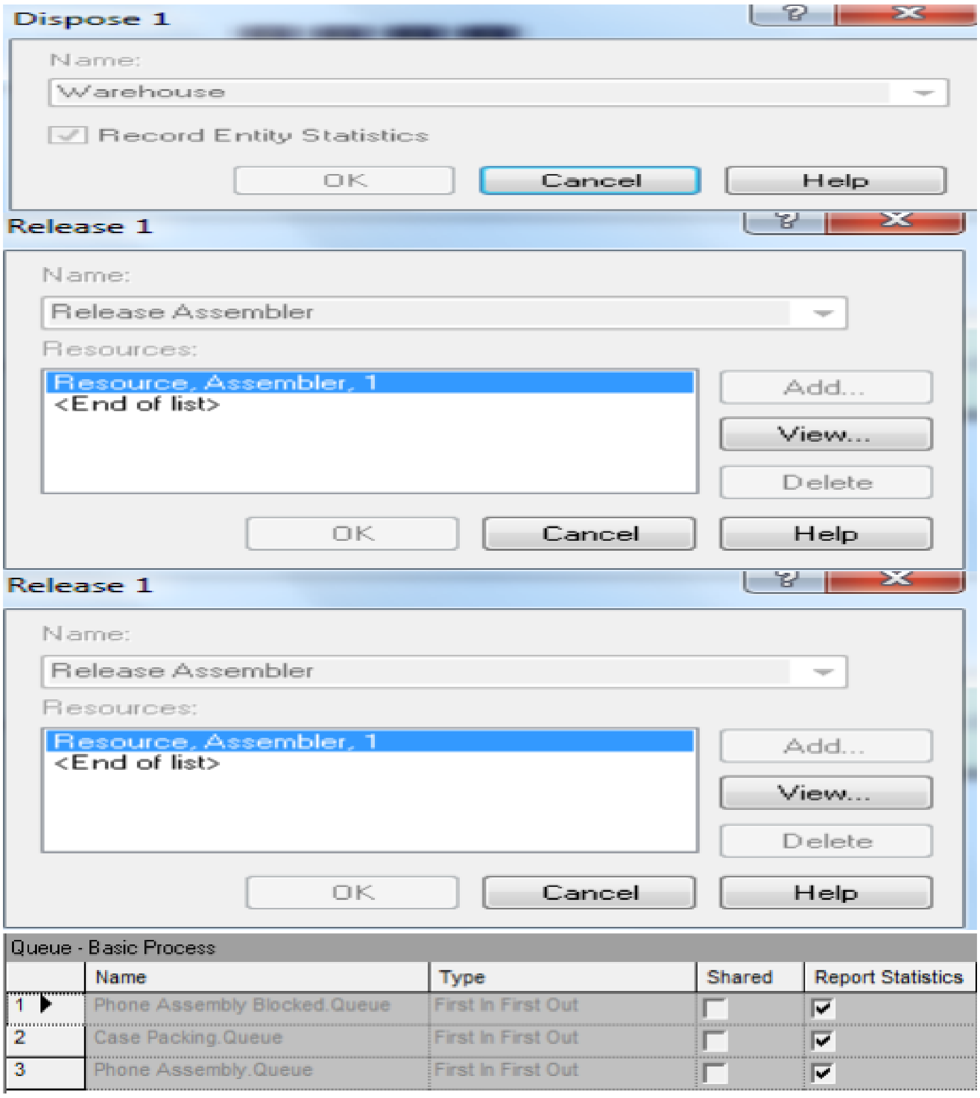


Figure 5.2 DES tool 1 (ARENA) used in Manufacturing design of the phone part production line

During the ARENA design planning the general structure of a phone part production line is set, primarily with a focus on the line's sequence of events. These are evaluated with respect to their modelling elements and their associated attributes.

During the experiment phase - the model of the body shop elements such as Create, meaning the beginning of creating the phone part process - you have an option to select the number of the entity, time between the arrival of a process, the types of the entity. In this case a constant was selected with value (one), the time is also specified in hours, minutes and seconds but we selected hour here because the machines are schedule to work for a shift of 8hrs, the number of the entity per arrival is set at one with the maximum arrival set at infinity in order to allow the completed process to arrive as soon as possible.

This is followed by processes defined as the Work Centre; the phone assembly was created with standard type of distribution, the seize delay logics in the distribution allowing the prioritising of the process. Resources are also chosen for the assembler; we can also add more resources – as much as required. The resources are held in the buffer (queue) until the resources are needed. The queue is used to place each process on hold until the next process is ready in order of arrangement.

In general, this case includes the initial generation of model data of the tools involved. The extraction of data from one DES model and the transforming of it with the data from other tool is accomplished through the Eclipse Atlas transformation environment which has been developed in this research. The use case is outlined in Figure 5.3; it involves the sharing of ARENA DES data into SIMUL8 DES and verse versa.

The development layer includes the setup of a basic structure of a phone part production line and describes the data contained and their basic hierarchical relations.

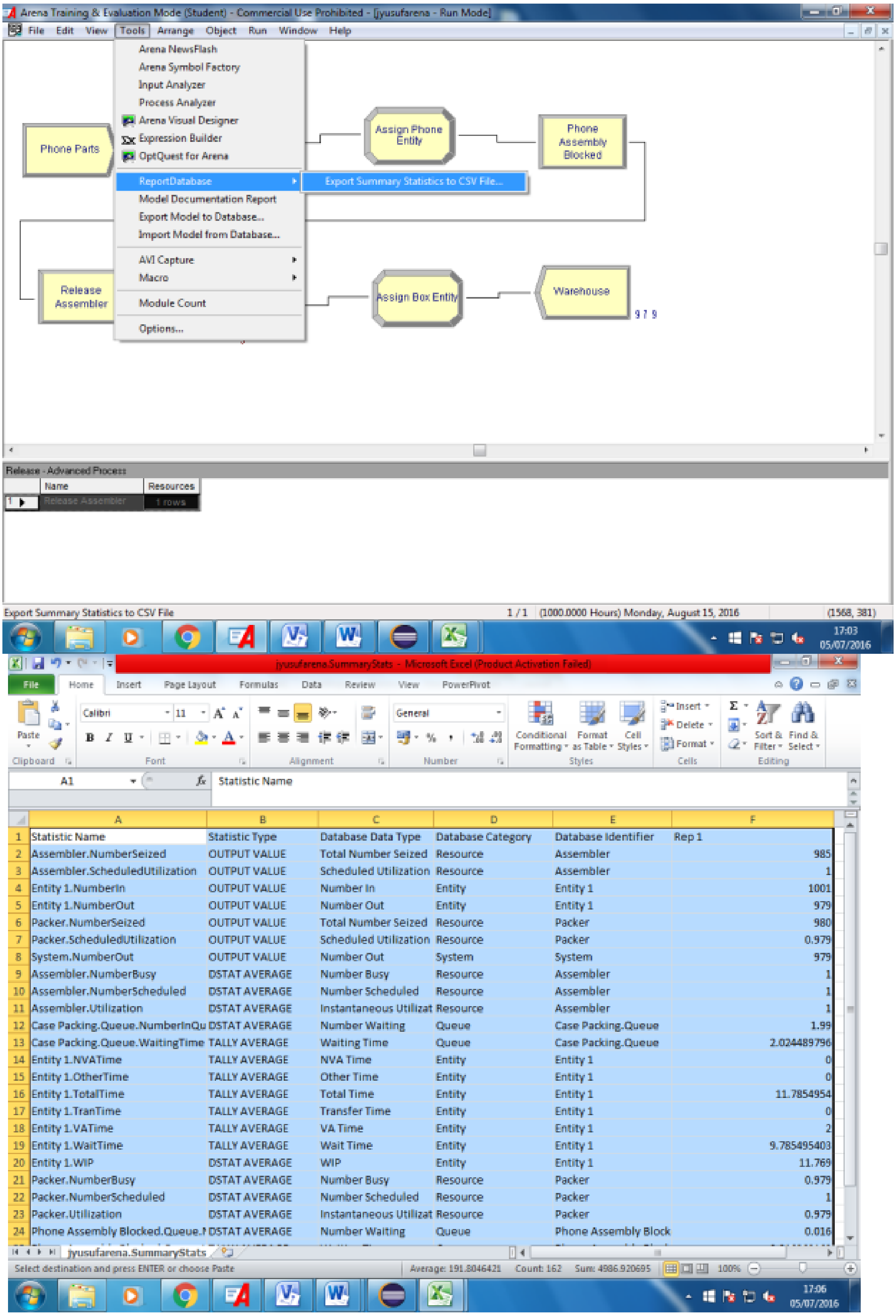


Figure 5.3 structure of a phone part production line (ARENA)

Figure 5.3 shows a screenshot from ARENA tool model of layout design of the phone part production line, a sample for this case study. The tool contains the entire model layout for one specific production line data and can be exported from the model in the access file which can be imported and consumed by another tool. As described in Figure 3.8, the file is first read by the model, and then the relevant information is filtered and transformed into standard format.

Then, the export summary saves the data exported by the model as an ARENA data on a network device where it can be accessed by SIMUL8 tool involved in the data sharing process, the import file for Arena is shown in APPENDIX 7. A corresponding folder structure is then set up in a temporary folder on the local hard disc. This folder structure corresponds to the relation of the major elements of a phone part production line.

In addition to saving on the file structure, it can be stored in any form necessary for the software tools being used. For the case study in this research, the DES tool provides the data as a character-separated value (CSV) file because most tools have an import interface for this file format. It is named and stored in the same position in the folder structure corresponding to the phone part production line, the imported data. The method of importing transformed data from both DES in Atlas Transformation Language User Interface is shown in Figure 5.5 and the results are highlighted in APPENDIX 3.

5.4 SIMUL8 to ARENA tool experiment

SIMUL8 is another DES tool, this involves the initial model data generation just like the ARENA tool by developing its model using the phone part production line data. Following the extraction of its data from the model it is then ready to be imported and consume by another tool such as ARENA, using the opportunity presented in the transformation process.

The use of this case is outlined in Figure 5.4; it involves the sharing of SIMUL8 data of the phone part production line.

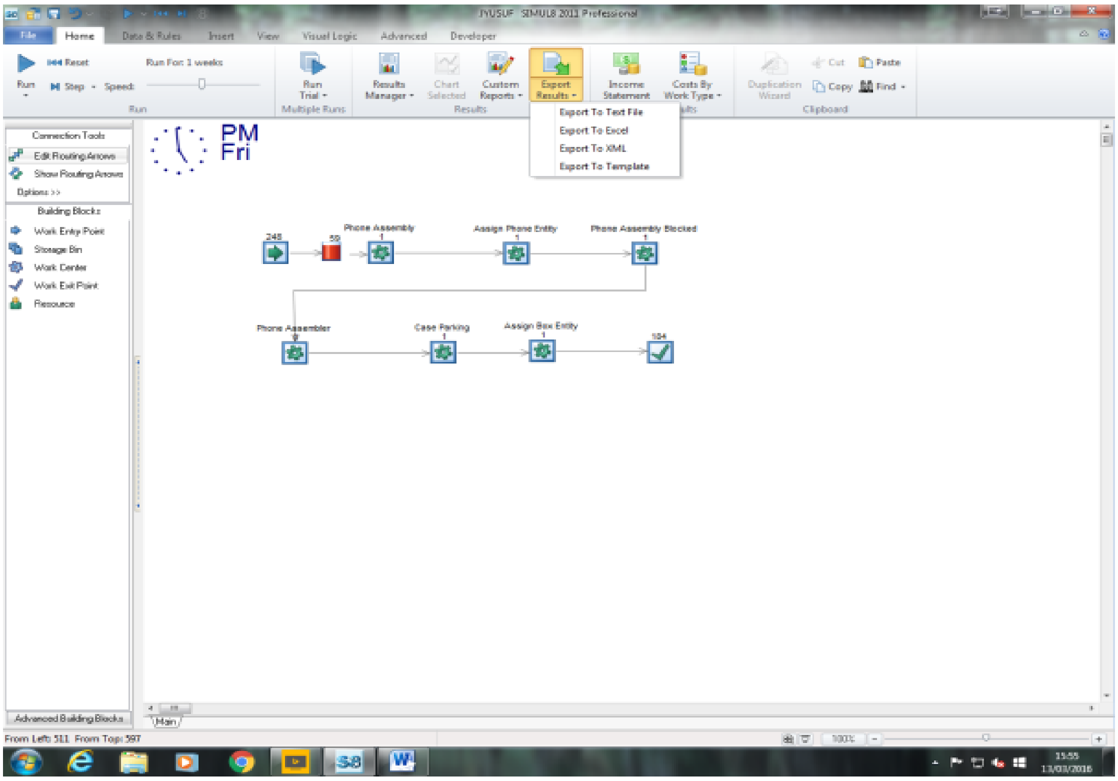


Figure 5.4 structure of a phone part production line (SIMUL8)

The SIMUL8 layout design includes the initial set up of a basic phone part production line and describes how data is represented and contained in the model.

Figure 5.4 shows a screenshot from a SIMUL8 model development for a phone part production line, another sample for DES in this research. The SIMUL8 model containing the phone part production line data can be exported from the model into XML or CSV as seen in the APPENDIX 8 for Simul8 import file which can be processed in MDDI and be ready to be consumed by the ARENA which is another case study in this research. The results are highlighted in APPENDIX 4.

A complete import by the SIMUL8 is shown in APPENDIX 8, then, the SIMUL8 model saves the data in its environment where it can be accessed by ARENA tool involved in the data sharing integration process. A corresponding folder structure is now set up in the temporary folder on the hard disk. This folder contains all data that corresponds with the relation of the elements of a phone part production line data.

5.5 Discussion of results

The case study presents the sequence initiation of the DES model prototype according to its model interaction and concepts. The main aim of this case study is to demonstrate that Model Driven Data Integration (MDDI) is applicable to DES data integration in a manufacturing production process. Moreover, this case study functions as a basis for the evaluation of the effort and the benefits of implementing the new MDDI.

The following subsection illustrates how a data import sequence from one DES is incorporated into another DES of a phone part production line in the case study. Building on this, the second and the third sub-sections introduce two use cases that demonstrate how data sharing has been achieved.

5.5.1 Implementation of Transformation

The research adopts merging models namely; SIMUL8 Process Framework Metamodel (SPFM) and ARENA Process Framework Metamodel (APFM) to define the relationships between the two DES model elements, this relationship of the metadata realises the data mapping and show how to move each of the data source to the target data.

The data merging model uses UML to express merging type and relationship of merging entities.

A model transformation rule was formed based on the merging model. The rule Create element _Create _ARENA is mapped from a source model to target DES tool model Work Entry Point_element. The rule has a set of {sources, targets} and the target has a set of {target element name, a set of attribute mapping}. Attribute mapping is expressed with arrows directing from source attributes to target attributes. Table 5.1 describes the description of each element.

Table 5.1 Transformation Rule (Hyeonsook et al., 2009)

Data Mapping Model	Transformation Rule
Model Element	If an element is connected to source model, generate a reference model source target. If attributes are connected to a source model element, generate a reference attributes source target.
Model to model relationship	Create a mapping as much as a source element is linked to the modeling element in the target model. Each data mapping from a source is transformed into the attributes of each connection between the target and the source in turns.

APPENDICES 5 and 6 define the Models Transformation results for SIMUL8 to ARENA and ARENA to SIMUL8 tool model, this is used to map source into a tool model. The results from the mapping and transformation are demonstrated in APPENDIX 4 and 5 respectively.

5.5.2 Import interface

This section describes the procedure for implementing the import interfaces for ARENA and SIMUL8 data. On this basis the labour-intensiveness of implementing a further import interface is evaluated. Both DES tools' file-based export is used for the import interface.

According to the integration architecture in Figure 3.8, based on MDDI modeling strategy, the import procedure is divided into three steps: first, read the data from the SIMUL8 to ARENA, second, ensure the filtration of data information and transfer it to the data model (see Figure 5.5). The import interface for Simul8 DES model data and ARENA DES model data was implemented.

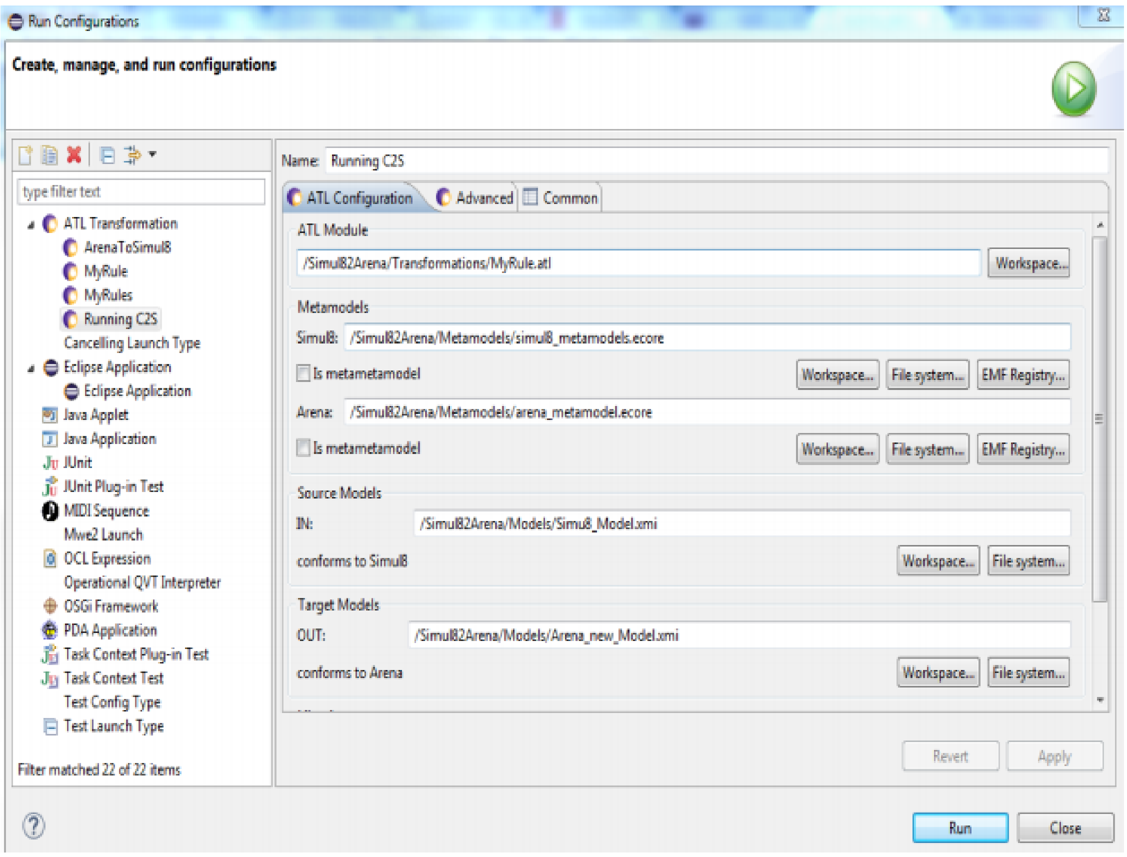


Figure 5.5: Atlas Transformation Language User Interface

To carry out the transformation the raw data from ARENA is imported and applied to SIMUL8 data model and vice-versa which offers and interfaces to export the phone part production line data. The extract result of the integration containing the new generic data is shown in APPENDIX 5 and 6.

5.6 Chapter summary

The characteristics of the implementation of this technique applied during the case study can be given as follows:

1. The implementation uses export interfaces. The advantage is that presently employed DES tools can be used and there is no need to purchase new DES tools and present manufacturing production design processes accordingly.
2. Storing the information of the DES tools' model data in text-based file formats or spreadsheet allows the various DES tools to keep track of the information sharing because they are able to control the file setup.
3. The significant investigation shows that the result of the transformation and integration capture all the required model and raw data which make the result complete. Furthermore, in this respect, there is a need to determine whether the mapping rule results are worth considering.

Here, in validating the result of the transformation, the level of correctness was proofing such that the mapping of the different model data successfully captures and included all the aspects of language definitions and their characteristics, this is demonstrated by showing the behavior of the models in term of capturing and accepting the manufacturing raw data.

The significance of the results also indicated that data size of the different DES models might not be the same, but this has been rectified by making sure that all the source model and target models are related and meant for the same process in the model. The approach of this research is a Model-Driven Data Integration (MDDI) method, meaning that the initial output is a set of mappings between DES data sources and one data source is mapped to a target data source using the mapping rules in an ATL open source transformation environment. This consequently allows input and output interfaces and furthermore allows for the transformation and integration to take effect.

Difficulties

Some of the difficulties in this research are searching for the accurate information due to the heterogeneous nature of discrete-event simulation data sources and the differentiation aspect: to know which data or information is relevant to whom, which chronological order will the data and information be processed and finally, in which form of the processed data and information be stored. But those limitations have also explained the capability of minimising them through the chosen technique in this research. The proposed solutions to those challenges have given a contribution and direction to achieving the research objectives of this research.

The result of this analysis provides not only a solid base for research but is also essential for setting up further advanced data integration solutions.

Chapter 6 - Evaluation Criteria

6.1 Introduction

In the previous Chapter the implementation of the new approach (MDDI) in a manufacturing production line data was demonstrated. The efficiency of this method is evaluated in this Chapter by discussing its impact on data sharing among different DES tools as they are applied in the manufacturing production line. The case studies are used to evaluate how well the data integration solution for each of the DES tools involved are able to face the challenges of data sharing.

6.2 Evaluation of the new method

Previously, the Model data integration strategy has been discussed and implemented using the manufacturing production process as a case study. Some factors such as nomenclature, differentiation and concept have been identified as the factors limiting the data sharing among different tools.

A manufacturing simple production line was used to develop and evaluate how the model driven data integration approach is able to deal with the challenges of data sharing among different DES models identified in the literature in section (**3.2.3**). They identified problems associated with the DES models such as heterogeneity, terminology etc.

The test in this research shows how the Model Driven Integration method is able to face the challenges of data integration to allow the DES tools to share their data models. The test cases are in many parts, such as identifying how the solution will react to the change in the models when adopted by another user, how the tools react to integration concept changes and finally, how it will react to changes when applying other DES models tools.

6.3 Use of the new method

6.3.1 Test case: Introducing New Data Models

The changes in model happen during the development of the DES model tool, meaning new entities can be added to the data model or change a process. The change refers to adding additional elements that can also be characterised by additional attributes.

Other changes such as an initial mistake during the simulation development phase, case test has been considered to react to this by simulating the test again based on the new data model and manufacturing production. Existing strategies such as top-down and bottom-up have no consideration for changes after the process has been completed, the limitation for this is that changes have to be made to each DES tools because the developers use independent data at it relate to a particular system or applications. And the existing strategy such as top-down method can only accommodate a few tools and has no way to differentiate between data relevant for multiple tools.

The MDDI has the ability to handle changes in DES data. Although, the necessary changes do not have much impact on the data sharing, but only for other DES tools such as ProModel, Simio e.t.c. Therefore, making the MDDI to have more advantages when compared to top-down and bottom-up approach.

6.3.2 Test 2: Applying other DES tools by another user

This explains how anyone who wants to adopt this approach, along with the possibility of adding other tools in the future and the possibility of effecting changes in the production life cycle after the initial transformation and integration, is the fundamental issue considered in this research. An ATL transformation and integration environment allow for any situation where the model transformation as designed can be made to share data from one DES tool to another.

Sharing another DES model data can only differ from any new tool introduced because there will be the need to create and define a new DES tools template, so that data can be extracted to the new template with sets of rules before it can be made to share its data.

The underlying DES tool currently adopted can be adopted for any manufacturing production line, but since manufacturing companies can decide to change or use different DES tools, there is a need to consider other DES tools, but the underlying advantage is that the tools can solve similar tasks and operate similarly to each other. This means that when a new DES tool is introduced, the new data models are likely to have similar information which will be stored in the template before linking them together to share data. However, this might compromise the integrity of the data sharing process because there might be more or less data produced and shared unnecessarily.

In summary, there is a need to make new DES tools to adapt to the rule for data sharing or be automated. However, the initial integration can remain untouched and the sharing of other new DES tools is very possible without hindrance because most modern DES tools have elements that are practically possible to share with other DES tools.

6.4 Correctness and completeness of results

The significant investigation in chapter 5 shows that the result of the transformation and integration capture all the required model and raw data which make the result complete and also in this respect, there is a need to determine whether the mapping rule results is worth considering.

Here, in validating the result of the transformation, the level of correctness was proofing such that the mapping of the different models data successfully captured and included all the aspect of language definitions and their characteristics, this is demonstrated by showing the behavior of the models in term of capturing and accepting the manufacturing raw data.

The significance of the results also indicated that the data size of the different DES models might not be the same, however this has been rectified by making sure that all the source model and target models are related and meant for the same process in the model. The approach of this research is a Model-Driven Data Integration (MDDI) method, meaning that the output is a set of mappings between DES sources and one data source is mapped to a target data source using the mapping rules in an ATL open source transformation environment. This consequently allows for the input and output interfaces and furthermore allows the transformation and integration to take effect.

Some of the limitations of this research are searching for the accurate information due to the heterogeneous nature of discrete-event simulation data sources, but those limitations are also capable of being minimised through the chosen technique in this research. The proposed solutions to those limitations have contributed towards achieving the research objectives of this research.

6.5 Chapter summary

This chapter presented the evalution of the new MDDI approach and its applications in a manufacturing production line system. The evalution identified the efficiency of the new method and its general impact on data sharing among different DES tools, this chapter also demonstrated how the new method can cope with introduction of other tools data that are not within the scope of current research. It also shows the tested cases on how new data models can be added and compared with existing strategies such as top-down and bottom-up. The correctness and completeness of the results of the transformation were also discussed, the result shows that the transformation was achieved and worth considering.

Chapter 7 - Conclusion

7.1 Introduction

This chapter draws the thesis to an end and summarises the contribution to knowledge resulting from this research. In addition, directions for future work are discussed. The research aim was to develop a model driven integration for data sharing among different discrete event simulation tools that take into account manufacturing production line data.

The above research aim of this development has been achieved.

Objective 1: **Development of two models representing the DES data models that take consideration of manufacturing production line data sources.**

Conclusion 1: This research achieved the objective through the use of artificial manufacturing production line data to develop two examples using discrete event simulation tools' models. The initial process was to understand the model structure of the system and the model data as described in the literature (section 3.5.1).

The aim was accomplished by using a simple production line of phone part production to understand how these tools have been used to model their production processes.

Most researchers and the applications for data integration have not fully considered integration of commercial DES data and much opinion has been focused on commercial simulation software customization making companies to always invest in expensive ready-made tools to share their data. However, the literature has linked this problem of data sharing among simulation tools to a general integration problem.

Objective 2: **Generate a generic representation of relationships among DES data sources with reference to production line and use DES modelling element (ME) and other attributes to obtain the model data.**

Conclusion 2: The DES was analysed to develop the model data which itself produced the data for input and the design component characteristics of the modeling elements and attributes that formed the data model as seen in Table 4.1 and APPENDIX 1 and 2, this allowed the definition the flow of data for manufacturing production line. In this research, all concepts relating the example models were evolved and their interaction ensures the efficiency of the use of the manufacturing data.

Objective 3: To identify through the literature, the concept and language definition and process interaction in different DES (e.g. ARENA and SIMUL8).

Conclusion 3: In terms of discrete event simulation, the importance of identifying how manufacturing data are presented is judged by the concept and language interactions. The Table **4.1** described the ME and attributes of the models and was defined as the concepts from various DES tools and harmonised into Process Framework Metamodel in Figure 4.8 and Figure 4.9 respectively and also shows language representation for visualising the concept associated with each model. The Metamodel class diagram defined the characteristics of how the manufacturing data can be shared.

Objective 4: To establish relationships between different DES through their modelling elements identified in Objective 1 and their process interaction.

Conclusion 4:

This objective has been achieved through the clinical analysis of process interaction and concept definition evolved in the UML class diagram. The interrelationship of the data model enables the combination of data from different sources and, furthermore, enables the users to have a unified view of the data.

Objective 5: To develop a mapping of using relationships between the DES tools to develop the Model Driven Data Integration (MDDI) technique that can enhance data sharing among DES data sources.

Conclusion 5:

This objective was effectively achieved, as integrating data from different sources can be time consuming and complex. This is because the process involves various data sources that are developed and designed by different vendors and used by various processes such as manufacturing process, which make it difficult for integration and even make it more difficult to share data.

In this research a model-driven data integration method was proposed to resolve the problems highlighted above to achieve the data integration for the DES data source. The method involved the incorporation and the utilisation of the metamodel for the data models. This research successfully applied and provides the transformation framework through MDDI to provide the integrated development to support the data reuse and sharing. The MDDI provides the support for the modeling of the metadata and the exchange of it between the tools, using model transformation rules.

The model transformation rules result is described in APPENDIX 5 and 6, one DES tool modeling elements is mapped into another tool modeling elements described as the target model data source; the rule is described in APPENDIX 5 and 6.

In conclusion, this research developed a method for data integration using MDDI with specific models, in this research the aim was to develop the Model Driven Data Integration (MDDI) technique that can enhance data sharing among DES data sources, this has been demonstrated through the mapping model that shows how this can be develop through the use of Eclipse model transformation environment using manufacturing data as a case study. In this research each data source was modeled with the UML to extract data from different data sources, and then the mapping model was developed to integrate all the data sources into one unified model. This research concluded that the model and the transformation can be adopted and be reused as a framework for the manufacturing production line and other applications that are required to integrate their data to enhance data sharing and to enhance good decision making. This research has also established ways that can be adopted to transform data, whereby the research adopted a model transformation technique using an open source Eclipse ATL model environment.

7.2 General Conclusions

The manufacturing system is growing in complexity and needs cooperation between the different simulation tools for how they use their operation processes. Subsequently, the MDDI in simulation system for the manufacturing industry must be able to react to this process and product driven environment.

In addition the MDDI make it possible for the DES tools to share their data since the immediate result can be applied to many tools until user requirements are met, this will make the manufacturing system able to shorten product time to market and increase their overall operational efficiency.

An MDDI was presented that enabled the efficient data integration between different DES models tools used by various manufacturing systems. The MDDI is a more efficient method of data integration because it aids the incorporation and utilisation of metadata through the data integration process. Providing an integrated standard development method, it also allows for the definition of relationships and interrelations between different models at a conceptual level, as well as providing an integrated development method for the supporting, interoperability, integration, portability, reusability and adaptability, and therefore, it can be adopted to solve any simulation tools for complex processes and systems.

Until now, there has been no existing commercial DES tools standard data sharing method. It is expected that other heterogeneous DES simulation data will continue to be complex and complicated, but with this research it is expected that more flexible solutions will be encouraged. More complex simulation tools will continue to make their way to market without considering how they can incorporate other tools' data, therefore making data integration an increasingly important and useful area for research.

7.3 Future work

This research has successfully proposed a Model Driven Data Integration (MDDI) technique that allows the use of different simulation tools by manufacturing systems to share their data. This research recommends that in the future:

1. Future work should incorporate other simulation tools to enhance the system integration in environments other than a manufacturing system at operation level (e.g. Making it easy to use shop floor data and compare it to the process design of the system) to improve the efficiency of the application and system.

2. A different approach should be taken to continue to resolve the issue of data reusability by undertaking research towards automatic data generation from different simulation tools using the MDDI.
3. The model characteristics should only be drawn primarily from two DES model data from literature and other tools should be required to be added within and outside manufacturing industry.
4. The MDDI developed includes models, Metamodel of the model, transformation rules, and data sharing. However, in the future there is a need to develop a template that can take data from the tools before making them available for sharing.

REFERENCES

Adikaram, B., Hussein, M. and Becker, T. (2015) Data Transformation Technique to Improve the Outlier Detection Power of Grubbs' Test for Data Expected to Follow Linear Relation. Journal of Applied Mathematics.

Akella, K., Benjamin, P., Malek, K. and Fernandes, R. (2005) An Ontology-Driven Framework for Process-Oriented Applications. Proceedings of the 2005 Winter Simulation Conference.

Amit, S., Cartic, R. and Christopher, T. (2005) Semantics for the semantic Web: the implicit, the formal and the powerful. Int'l Journal on Semantic Web & Information Systems, Vol.1, Issue 1, pp. 1-18.

Arief, L. and Speirs, N. (2000) A UML Tool for an Automatic Generation of Simulation Programs.Proceedings of the Second International Workshop on Software and Performance.

Babulak, E., and Ming, W. (2012) Discrete Event Simulation: State of the Art. Discrete Event Simulation: State Of The Art.

Bengtsson, N., Shao, G. and Johansson, B. (2009) Input Data Management Methodology For Discrete Event Simulation. Proceedings of The 2009 Winter Simulation Conference.

Bergman, M. (2014) Big structure and data interoperability. Adaptive Information.

Bergman, M. (2006) Sources and classification of semantic heterogeneities. Adaptive Information.

Bishr, Y. A. (2008) Overcoming the semantic and other barriers to GIS interoperability. International Journal of Geographical Information Science, Vol.12, No 4, pp. 229–314.

Bloom, H. (2004) Technical Program Description Systems Integration for Manufacturing. NISTIR 5476, NIST Gaithersburg.

Bock, C. (2003) UML 2 Activities and Action Models.Journal of Object Technology, Vol. 2, No 4.

Brickley, D. and Guha, R. (2004) RDF Vocabulary Description Language 1.0: RDF Schema. http://www.w3.org/TR/rdf-schema.

Bock, C. (2003) UML 2 Activity and Action Models.Journal of Object Technology, 2(4).

Bruni, P. et al., (2003) Data federation with IBM DB2 Information Integrator.V8.1 http://www.redbooks.ibm.com/redbooks/pdfs/sg247052.pdf.

Calvanese, D., Giacomo, G. and Lembo, D. (2005) Inconsistency tolerance in P2P data integration: an epistemic logic approach. In Proc. of the Tenth International Symposium on Database Programming Languages, pp. 90–105.

Carlos, R. Inma, H., David, R. and Rafael, C. (2013) MostoDE: A tool to exchange data amongst semantic-web ontologies. Journal of Systems and Software, Vol. 86, Issue 6, pp. 1517–1529.

Chae, S., Hyun, S. and Young, C. (2014) Model Transformation Rule for generating Automatic Database Schema of Business Process Framework. International Journal of Software Engineering and Its Applications, Vol. 8, Issue 3, pp. 47-54.

Cruz, I. and Xiao, H. (2003) Using a Layered Approach for Interoperability on the Semantic Web. In Proceedings of the 4th International Conference on Web Information Systems Engineering, pp. 221-232.

Davenport, T. H. and Harris,. J. (2007) Competing on analytics: The new science of winning, Boston: Harvard Business School Press.

Djamel, A. B. (2014) Facilitating the specification of WSMO ontology using model-driven development.International Journal of Metadata, Semantics and Ontologies, Vol. 9, Issue 2.

Douglas , S. (2006) Model Driven Engineering. *IEE Computer, Vol.*39, Issue 2, pp. 25-31.

Eric, M. (2008) An Introduction to the Resource Description Framework, Dublin: D-Lib Magazine.

Ericsson, U. (2005) Diffusion of Discrete Event Simulation in Swedish Industry One way to an Increased Understanding, PhD, Gothenburg: ISBN 91-7291-577-3.

Fazli, E., Fuxman, A. and Miller, R. (2005) ConQuer: Efficient management of inconsistent databases ACM SIGMOD International Conference on. pp. 155–166.

Fishwick, P. A. and Miller, J. (2004) Investigating Ontologies for Simulation Modeling.Simulation Symposium.

Garg, D. and Tyagi, M. (2012) Comparative Analysis of Dynamic Graph Techniques and Data Structure. International Journal of Computer Applications, Vol.45, Issue 5, pp. 0975-8887.

Greco, G. et al. (2005) The infomix system for advanced integration of incomplete and inconsistent data.SIGMOD Conference.

Haas, L. (2003) A researcher's dream. DB2 Magazine Vol.8, Issue 3, pp. 34-40.

Hao, Q. et al., (2006) Agent-based collaborative product design engineering:An industrial case study. *Computers in Industry*, pp. 26-38.

Hauge, J. W. and Paige, K. N. (2002) Learning SIMUL8: The Complete Guide, Bethlingham: Plain Vu Publishers.

Huiyong, X. and Isabel, C. (2004) Integrating and Exchanging XML Data using Ontologies. 8th International Database Engineering & Applications Symposium, pp. 217-226.

Hyeonsook, K., Ying, Z., Samia, O. and Tony, C. (2009) A Case Study on Model Driven Data Integration for Data Centric Software Development. DSMM'09.

Hyun, S. S., Jae, K. S. and Robert, K. (2013) SMTL Oriented Model Transformation Mechanism for Heterogeneous Smart Mobile Models. International Journal of Software Engineering and Its Applications, Vol.7, Issue 3.

IBM websphere information integrator. (2005) Accessing and integrating diverse data for the on.IBM White Paper.

Jianbo, L., Yiping, Y. Wenjie, T. and Feng, Z. (2015) Research on the Development Approach for Reusable Model in Parallel Discrete Event Simulation.Discrete Dynamics in Nature and Society.

Jing, D., Sheng, Y. and Kang, Z. (2014) A Model Transformation Approach for Design Pattern Evolutions. International Journal of Software Engineering and Its Applications, pp. 40-58.

Joachim, H. and Charnyote, P. (2010) A classification scheme for semantic and schematic heterogeneities in XML data sources, Florida: Technical Report TR00-004.

Johansson, M. et al., (2007) A Test Implementation of the Core Manufacturing Simulation Data Specification. Proceedings of the 2007 Winter Simulation Conference, eds, pp. 1673-1681.

Jones, T., Wysk, R. A. and Son, Y. (2003) A Component Based Simulation Modeling from Neutral Component Libraries. Computers & Industrial Engineering, Volume 45, pp. 141-165.

Kang, P. et al., (2015) Process Control Parameters Evaluation Using Discrete Event Simulation for Business Process Optimization. Proceedings of the 64th IWCS Conference, pp. 567-577.

Kelton, D., Sadowski, P. R. and Sturock, T. (2007) Simulation with Arena, New York: McGraw-Hill.

Kirchner, L. and Jung, J. (2007) Framework for the Evaluation of Meta-Modelling Tools. The Electronic Journal Information Systems Evaluation, Vol. 10, Issue 1, pp. 65 - 72.

Kleppe, A. and Warmer, J. (2003) MDA Explained. Reading: Addison-Wesley.

Kolaitis, P. G. & Fagin, R. (2003) Data exchange: Getting to the core. In Proceedings of the Twentysecond ACM SIGACT SIGMOD SIGART Symposium on Principles. pp. 90-91.

Kondylakis, H. and Plexousakis, D. (2011) Exelixis: Evolving Ontology-Based Data Integration System. *Proceedings of the 2011 ACM SIGMOD International* , pp. 1283-1286.

Lafortune, S. and Cassandras, C. (2009) Introduction to Discrete Event Systems. Kluwer: s.n.

Lenzerini, M. (2002) Data integration: A theoretical perspective. In Proceedings of the TweIn Proceedings of the Twentyfirst ACM SIGACT SIGMOD SIGART Symposium on Principles of Database Systems, pp. 233–246.

Leong, S. (2006) A Core Manufacturing Simulation Data Information Model for Manufacturing Applications, Gaithersburg: Manufacturing Systems Integration Division National Institute of Standards and Technology.

Lina, T. and Robertas, D. (2009) Characteristics of Domain Ontologies for Web Based Learning and their Application for Quality. Institute of Mathematics and Informatics, Vol. 8, Issue 1, pp. 131-152.

Low, Y. H. et al., (2003) Implementation issues for shared state in HLA-based distributed simulation. Netherlands, Eur. Simulation Symp.

Michael, I. (2009) Overview of the Manufacturing Engineering Toolkit Prototype. Manufacturing Systems Integration Division, Manufacturing Engineering Lab.

Mira, J. and Fernández, C. (2003) Two Examples of Deterministic versus Stochastic Modeling of Chemical Reactions. Journal of Engineering, Vol. 80, Issue 12, pp. 1488.

Moser T, Zoitl, A. & Waltersdorfer, F., 2010. Version management conflict detection across homogenous engineering data model. *IEEE,* pp. 928-935.

Nance, R. E. and Overstreet, C. M. (2005) A Specification Language to Assist in Analysis of Discrete Event Simulation Models.Communications of the ACM, 28(2).

Narain, S. (2001) An Axiomatic Basis for General Discrete-event Modeling. Proceedings of the 23rd Winter Simulation Conference.

Nehme, C. and Cummings, L. (2009) Modeling Human Supervisory Control in Heterogeneous Unmanned Vehicle. 9th Conference on Performance Metrics for Intelligent, Vol. 6, Issue 3, pp. 381-401.

Object Management Group (OMG). (2009) Unified Modeling Language. Specification 2.2: s.n.

Pace, D. K. (2006) Conceptual Model Development for C4ISR Simulations.Fifth International Command and Control Research and Technology Symposium.

Pérez, d. L., Mendling, J. and Zdun, U. (2005) Towards Semantic Integration of XML based Business Process Models. Proceedings of the WM2005 Professional Knowledge

Management - Experiences and Visions. Semantic Model Integration Workshop (SMI2005) as part of the 3rd Conference Professional Knowledge Management, pp. 513-517.

Pidd, M. (2002) Simulation Worldviews – So What? Proceedings of the 2004 Winter Simulation Conference.

Pilone, D. (2005) UML 2.0 in a Nutshell. s.l.:O'Reilly.

Praehofer, H. and Pree, D. (2003) Visual modeling of DEVS-based multiformalism systems based on higraphs. S.l. Proceedings of the 25th Winter Simulation Conference.

Praehofer, H., Zeigler, B. P. and Kim, T. G. (2000) Theory of Modeling and Simulation. 2nd ed. San Diego: Academic Press.

Reichenthal, S. W. and Gustavson, P. (2003) Manufacturing BOMs with SRML for Process Oriented Federations. Proceedings of the fall 2003 Simulation Interoperability workshop.

Sandanayake, G., Oduoza, C. F. and Proverbs, D. (2008) A systematic Modelling and Simulation approach. *Robotics and computer -Integrated Manufacturing,* Vol. 24, Issue 6, pp. 735-743.

Seila, A. F. (2005) The Case for a Standard Model Description for Simulation.International Journal of Simulation and Process Modeling, Vol.1, Issue 2.

Siebers, P. O. et al., (2010) Discrete-event simulation is dead, long live agent-based simulation. *Agent-Based Simulation!'. Journal of Simulation,* Vol. 4, Issue 3, pp. 204-210.

Silver, G. A. and Miller, J. A. (2006) Ontology Based Representations Of Simulation Models Following The Process Interaction World View. Proceedings of the 2006 Winter Simulation Conference.

Simeone, M., Guiliano, J., Kooper, R. and Bajcsy, P. (2011) Digging into data using new collaborative infrastructures supporting humanities–based computer science research. *Humanities–based computer science research,*Vol. 16, Issue(5).

Skoogh, A. (2012) Methods for Input Data Management.Reducing the Time-Consumption in Discrete Event Simulation.

Stephan, P. and Jacob, K., (2006) Addressing the problems with life-science databases for traditional uses and systems biology. Nat Rev Genet, Vol.7, Issue 6, pp. 482-488.

Storrle, H. and Hausmann, J. (2005) Towards a Formal Semantics of UML 2.0 Activities. Software Engineering, pp. 117-128.

Sudra, R., Taylor, J. E. and Janahan, T. (2000) Distributed supply chain management in GRIDS. Winter Simulation Conference Orlando, FL, pp. 356-361.

Sujansky, W. (2001) Heterogeneous database integration in biomedicine. J Biomed Inform, Vol. 34, Issue 4, pp. 285–298.

Swain, J. (2014) Simulation Reloaded OR/MS Today http://www.gridbus.org/ papers simulationtaxonomy.pdf.

Swee, L., Tina, L. and Frank, R. (2007) A Core Manufacturing Simulation Data Information Model for Manufacturing Applications. National Institute of Standards and Technology: Manufacturing Systems Integration Division, pp. MD 20899-8260.

Swets, N. and Bapat, V. (2000) The Arena Product Family: Enterprise Modeling Solutions. Proceedings of the 2000 Winter Simulation Conference.

Trybula, W. (2004) Building simulation models without data. *IEEE International conference of systems, Man and Cybernetics*, pp. 209-214.

Vassiliadis, P., Simitsis, A. and Skiadopoulos, S. (2002) Conceptual Modeling for ETL Processes.OLAP'02.

Virginija, U. and Rimantas, B. (2011) Ontology-based Foundations for Data Integration.The First International Conference on Business Intelligence and Technology.

Wan-Teh, C., Soonhoi, H. and Edward A. (2007) Heterogeneous Simulation-Mixing Discrete-Event Models with Dataflow.Journal of VLSI Signal Processing, Vol. 15, pp. 127–144.

Youcef, G. and Abdelhabib, B. (2012) Ontology and automatic code generation on modeling and simulation. 6th International Conference on Sciences of Electronics, IEEE.

Zeigle, P., Praehofer, H. and Kim, T. G. (2000) Theory of Modeling and Simulation. 2nd ed. San Diego: Academic Press.

APPENDICES

APPENDIX 1: SIMUL8 Source File

Source Table/File Name	Source filed name	Source filed name	Source filed name	Source filed name	Source filed name
Packages	Modeling element type	ME Name	Attributes	Associate attributes	Associate attributes
SIMUL8	Work Entry point	Imput work item type	Number of work item entered	Distribution	Average, exponential, triangulation
	Work Entry point	First entity arrive	Inter-arrival time	value	Average
	Work Entry point	Entity arrival	First at start time	Receive the first entity	timing Entity enter work center
SIMUL8	Modeling element type	ME Name	Attributes	Associate attributes	Associate attributes
	Work center	Work centre type	Time in system	Time limit	Hour minutes, seconds
	Work center	Number of work item	percentage of time	change over	Batch size
SIMUL8	Modeling element type	ME Name	Attributes	Associate attributes	Associate attributes
	Storage Bins	Queue for work centre	Number of work item in storage	Minimum,average, maximum	LIFO
	Storage Bins	Storage Area	Capacity	maximum number of item	High Volume
	Storage Bins	Storage Area	Queue time	Minimum average, maximum	Shelf Life
	Storage Bins	Storage Area	Queue time within limit	Time limit	Segregate Results

SIMUL8	Modeling element type	ME Name	Attributes	Associate attributes	Associate attributes
	Resources	Resources Name	Number of resources available	utilization	cost
	Resources	Resources Name	Shifts Dependent	Resources available on schedule	time
	Resources	Resources Name	Pool resources	specifies that resource	Usage
SIMUL8	Modeling element type	ME Name	Attributes	Associate attributes	Associate attributes
	Work Exit Points	work completed	Time in system	Time in system within limit	
	Work Exit Points	Collect results	Travel time	clock	
	Work Exit Points		warming up period	Terminate Run	
	Work Exit Points		hours per day	result collection period	
	Work Exit Points		Run finish	halt simulation	
	Work Exit Points		clock properties	simulation speed	
	Work Exit Points		start time	Simulation run time	
	Work Exit Points	KPI	Completed products count	Shift work pattern	
	Work Exit Points		Minimum	Equipment utilization	
	Work Exit Points		maximum	Cycle times	
	Work Exit Points		standard deviation	Work-in-progress	

APPENDIX 2: ARENA Source File

Target Table/File Name	Target filed name	Target filed name	Target filed name	Target filed name	Target filed name
Packages	Modeling elemnt type	ME Name	Attributes	Associate attributes	Associate attributes
ARENA	Create Module	Imput entity type	First creation	Type	Constant, randon(exponential) , average
	Create Module	Entity per arrival	Time between arrival	value	units
	Create Module	entity arrival	Unlimited arrival	First creation	hours
ARENA	Modeling element type	ME Name	Attributes	Associate attributes	Associate attributes
	Proccess	Process type	Delay type	Unit	hours
	Proccess	Logic	Minimum	value	maximum
ARENA	Modeling elemnt type	ME Name	Attributes	Associate attributes	Associate attributes
	Queues	Queu for Process	Lowest and higher attributes first	Shared	LIFO
	Queues	Storage Area1	Capacity	Report Statistics	High Volume
	Queues	Storage Area	Queue time	Minimum,average, maximum	Shelf Life
	Queues	Storage Area	Queue time withing limit	Time limit	Segregate Results
ARENA	Modeling element type	ME Name	Attributes	Associate attributes	Associate attributes
	Resources	Resources Name	Type of resources available	Capacity	Busy/Hour

	Resources	Resources Name	schedule rule	Quantity	Idle/Hour
	Resources	Resources Name	select resources	input resources	Per Use
ARENA	Modeling element type	ME Name	Attributes	Associate attributes	Associate attributes
	Dispose Module	Work completed	Number of replication	replication length	
	Dispose Module	Collect results	warming up period	time units	
	Dispose Module		replication length	Terminate Run	
	Dispose Module		hous per day	Base time Units	
	Dispose Module		Average Time in system	run setup	
	Dispose Module		Record entity statitics	clock properties	
	Dispose Module		start time	Time in second simulation	
	Dispose Module	KPI	Completed products count	Shift work pertern	
	Dispose Module		Minimum	Equipment utilization	
	Dispose Module		Maximum	Cycle times	
	Dispose Module		standard deviation	Work-in-progress	

APPENDIX 3: Transformed DES (SIMUL8) Data

Work Type ID (Simul8)	ME Type	ME Name	Succeeding ME	Processing Time	Processing Time Distribution	Change Over Time	Change Over Time Distribution	MTBF (Machine Failure: How Often)	MTBF Distribution	MTTR (Machine Failure: Time to Fix)	MTTR Distribution	Usage Cost Per Unit Distribution	Number of Resources	Travel Time	Travel Time Distribution	Cost Per Unit Time Distribution	Queue Capacity	Shelf Life	Start up Items (Number)	Min Waiting Time	Interarrival Time	Interarrival Time Distribution	Batch Size	Cost Per Unit Distribution
Phone part design	Work Entry Point	Order In	Order Queue																		1 Week - 4 Months	Range	1 to 20	Range
	Storage Bin	Order Queue	Phone part														10	Infinite	1	N/a				
	Work Center	Phone assembly	Assembler	10 - 48 Hours	Range	N/a	N/a	N/a	N/a	N/a	N/a		1	0	Average	Average								
	Work Center	Assign phone entity	Product Data	2 - 8 Hours	Exponential	1 Day - 2 Weeks	Range	2 Months	Average	1 - 4 Hours	Range	Average	1	0	Average	Average								
	Work Center	Phone assembly blocked	Hold	6 - 24 Hours	Range	1 Day - 2 Weeks	Range	2 Months	Average	1 - 4 Hours	Range	Average	1	0	Average	Average								
	Work Center	Release assembler	Release	10 - 30 Hours	Average	1 Day - 2 Months	Range	4 Months	Average	1 Day - 1 Week	Range	Average	3	1 Min	Average	Average								
	Work Center	Case Parking	Queue for Technicians	2 - 6 Hours	Range	N/a	N/a	1 Months	Average	1 - 4 Hours	Range	Average	1	1 Min	Average	Average								
	Work Center	Assign Box entity	Quality Control Assesment	4- 8 Hours	Range	N/a	N/a	7 Months	Average	1 - 4 Hours	Range	Average	1	1 Min	Average	Average								
	Work Center	Refers to Technicians	Fixing Process	20Hours	Average	N/a	N/a	N/a	N/a	N/a	N/a		1	0	Average	Average								
	Work Center	Completed Cases	Work Complete/Exit	6 - 14 Hours	Range	1 Day - 2 Weeks	Range	8 Months	Average	1 - 4 Hours	Range	Average	1	0	Average	Average								

Work Center
Operator
Storage Area (Queue)
Work entry Point
Work Complete/Exit
Modelling element
Associate attributes

APPENDIX 4: Transformed DES (ARENA) Data

Work Type ID(Arena)	ME Type	ME Name	Succeeding ME	Time between arrival (Units)	Type	Change Over Time	Change Over Time Distribution	MTBF (Machine Failure: How Often)	MTBF Distribution	MTTR (Machine Failure: Time to Fix)	MTTR Distribution	Usage Cost Per Unit Distribution	Number of Resources	Travel Time	Travel Time Distribution	Cost Per Unit Time Distribution	Queue Capacity	Shelf Life	Start up Items (Number)	Min Waiting Time	Interarrival Time	Interarrival Time Distribution	Batch Size	Cost Per Unit Distribution
Phone part design	Create Module	Order In	Order Queue																		Infinites	Range	1 to 20	Range
	Queue Spreadsheet	Order Queue	Phone part														10	Infinite	1	N/a				
	Process Module	Phone assembly	Assembler	600 - 2880 minutes	random (expo)	N/a	N/a	N/a	N/a	N/a	N/a		1	0	Average	Average								
	Process Module	Assign phone entity	Product Data	120 - 480 minutes	random (expo)	1 Day - 2 Weeks	Range	2 Months	Average	1 - 4 Hours	Range	Average	1	0	Average	Average								
	Process Module	Phone assembly blocked	seize delay	360- 1440 minutes	Triangular	1 Day - 2 Weeks	Range	2 Months	Average	1 - 4 Hours	Range	Average	1	0	Average	Average								
	Process Module	Release assembler	Hold/ scan for condition	600 - 1800 minutes	random (expo)	1 Day - 2 Months	Range	4 Months	Average	1 Day - 1 Week	Range	Average	3	1 Min	Average	Average								
	Process Module	Case Parking	Release	120 -360 minutes	Trangular	N/a	N/a	1 Months	Average	1 - 4 Hours	Range	Average	1	1 Min	Average	Average								
	Process Module	Assign Box entity	Assign Entity, picture, pic	240- 480 minutes	random (expo)	N/a	N/a	7 Months	Average	1 - 4 Hours	Range	Average	1	1 Min	Average	Average								
	Process Module	Refers to Technicians	Fixing Process	1200 minutes	Average	N/a	N/a	N/a	N/a	N/a	N/a		1	0	Average	Average								
	Work Center	Completed Cases	Dispose Module	360 - 840 Hours	Range	1 Day - 2 Weeks	Range	8 Months	Average	1 - 4 Hours	Range	Average	1	0	Average	Average								

Process Module
Operator
Order(Queue)
Create Module
Work Complete
Modelling element
Associate attributes

APPENDIX 5: SIMUL8 to ARENA Transformation Mapping Rules

SIMUL8 to ARENA Transformation Results.

```
-- @path Simul8=/model.simul8/model/simul8.ecore
-- @path Arena=/model.arena/model/arena.ecore
-- @nsURI Simul8= http://model.simul8/1.0
-- @nsURI Arena= http://model.arena/1.0

module Simul8ToArenaMapping;
create OUT : Arena from IN : Simul8;

rule WorkEntryPoint2Create {
	from
		workEntryPoint : Simul8!WorkEntryPoint
	to
		Create : Arena!Create (
			name <- workEntryPoint.name,
			entityPerArrival <- workEntryPoint.firstAtStartTimes,
			maxArrivals <- workEntryPoint.unlimitedArrivals,
			timeUnit <- workEntryPoint.timeUnit,
			timeBetweenArrivals <- workEntryPoint.distribution,
			queues		<-		workEntryPoint.routingOut->collect
(q|thisModule.StorageBin2Queue(q))
		)
}

rule WorkCenter2Process {
	from
		workCenter : Simul8!WorkCenter
	to
		process: Arena!Process (
			name <- workCenter.name,
			reportStatistics <- workCenter.highVolume,
			delayType <- workCenter.distribution,
			inputQueues		<-		workCenter.routingIn->collect
(q|thisModule.inputBuffer2Queue(q)),
			ouputQueues		<-		workCenter.routingOut->collect
(q|thisModule.OutputBuffer2Queue(q))
		)
}

rule Resource2Resource {
	from
		resource : Simul8!Resource
	to
		resourceA : Arena!Resources  (
			name <- resource.name,
			busyHour <- resource.cost,
```

```
                    idleHour <- resource.time,
                    perUse <- resource.usage,
                    type <- resource.shiftDependence,
                    reportStatistic <- resource.utilization
            )
}

rule ExistPoint2Dispose {
    from
            workexistPoint : Simul8!WorkExistPoint
    to
            dispose : Arena!Dispose (
                    name<-workexistPoint.name,
                    workCompleted<-workexistPoint.workCompleted,
                    collectResult<-workexistPoint.collectResult,
                    haltSimulation <-workexistPoint.haltModelAtLimit,
                    recordStatistics <- workexistPoint.segregateResult,
                    terminateRun <-  workexistPoint.highvolume
            )
}

rule SimulationClock2RunSetUp {
    from
            simulClock : Simul8!SimulationClock
    to
            runSetup : Arena!RunSetup (
                    simulate <- simulClock.run,
                    pauseAfterWarning <- simulClock.resetToSetup,
                    terminationCondition <- simulClock.beepOnCompletion,
                    warmingUpPeriod <- simulClock.warmUpPeriod,
                    resultCollectionPeriod <- simulClock.resultCollectionPeriod

            )
}

rule simKpid2AreKpi {
    from
            simKpi :Simul8!Kpid
    to
            areKpi : Arena!Kpi (
                    CompletedProductCount <- simKpi.CompletedProductCount,
                    equipmentUtilization <- simKpi.equipmentUtilization,
                    cycleTime <- simKpi.cycleTime,
                    leadTime <- simKpi.leadTime,
                    workInProgress <- simKpi.workInProgress
            )
}

rule ModelToModel {
    from
```

```
		SimModel : Simul8!Model
	to
		ArenModel : Arena!Model (
			description <- SimModel.description,
			name <- SimModel.name,
			timeUnit <- SimModel.timeUnits

		)
}


lazyrule RunTimeSimulationClock2RunningTimeRunSetUp {
	from
		--runTimeSimul                                                    :
Simul8!SimulationClock(runTimeSimul.oclIsTypeOf(runTimeSimul))
		runTimeSimul : Simul8!RunningTime
	to
		timerunSetUp :  Arena!RunningTime  (
			startDayAndTime <- runTimeSimul.startTimeAtEachDay,
			hourPerDay <- runTimeSimul.timeInEachDay
		)
}

lazyrule SimulationSpeedClock2RunSpeedRunSetUp {
	from
		 simulSpeed : Simul8!RunSpeed
	to
		RunSpeedSetUp : Arena!SimulationSpeed (
			numberOfReplication <- simulSpeed.speedControl0-100,
			replicationLenght <- simulSpeed.typicaMinutesPerRealSecond
		)
}

lazyrule ClockPropertiesClock2SClockPropertiesRunSetUp {
	from
		ClckPperties : Simul8!ClockProperties
	to
		ClkPperties : Arena!ClockProperties (
			timeUnit <- ClckPperties.TimeUnit
		)
}

lazyrule StorageBin2Queue {
	from
		storage:Simul8!StorageBin
	to
		queue : Arena!Queue (
			name <- storage.name,
			reportStatistics <- storage.isHighVolume
		)
```

```
}

lazyrule inputBuffer2Queue {
	from
		storage : Simul8!StorageBin (storage.oclIsKindOf(Simul8!InputBuffer))--
InputBuffer StorageBin
	to
		queue : Arena!Queue (
			name <- storage.name,
			reportStatistics <- storage.isHighVolume
		)
}

lazyrule OutputBuffer2Queue {
	from
		storage : Simul8!StorageBin (storage.oclIsKindOf(Simul8!OutputBuffer))
	to
		queue : Arena!Queue (
			name <- storage.name,
			reportStatistics <- storage.isHighVolume
		)
}

lazyrule TriangularDistribution2TriangularArrivalType {
	from
		--triDist : Simul8!Triangular(triDist.oclIsKindOf(Simul8!Distribution))--
triangular distribution
		triDist : Simul8!Triangular
	to
		arivalTri : Arena!Triangular (
			allocation <- triDist.mode,
			maximum <-   triDist.upper,
			minimum <- triDist.lower,
			value <- triDist.mode
		)
}

lazyrule FixedDistribution2FixedCapacityArrivalType {
	from

		--fixDist :Simul8!Fixed (fixDist.oclIsTypeOf(Simul8!Distribution)) --fixed
distribution
		fixDist:Simul8!Fixed
	to
		arivalFix : Arena!FixedCapacity (
			capacity <- fixDist.value

		)
}
```

```
lazyrule RoundedUniformDistribution2UniformArrivalType {
	from
	    --uniDist : Simul8!Distribution(uniDist.oclIsTypeOf())
	          formDist : Simul8!RoundedUniform
	to
		arivalUni : Arena!Uniform  (
			minimum <-formDist.lowerBound,
			Maximum<-formDist.upperBound
		)
}

lazyrule ExponentialDistribution2RandomArrivalType {

	from
	--expoDis:Simul8!Exponential(expoDis.oclIsTypeOf(Simul8!Distribution))
		expoDis : Simul8!Exponential
	to
		randomArival : Arena!Random (
			value <- expoDis.vaule,
			firstCreation <- expoDis.average
		)
}

lazyrule LogNormlDistribution2NormalArrivalType {
	from
		--lodNorm : Simul8!LogNormal(lodNorm.oclIsTypeOf(Simul8!Distribution))
		lodNormDist: Simul8!LogNormal
	to
		normArival : Arena!Normal (
			value <- lodNormDist.value,
			stdDeviation <- lodNormDist.stdDeviation,
			units <- lodNormDist.unit,
			allocation <- lodNormDist.average
		)
}

lazyrule ScheduleRulelResourceType2ShiftResource {
	from
		shiftRes: Simul8!Shift
	to
		schedRes : Arena!ScheduleRes (
			RULE <- shiftRes.behaviour,
			capacity <- shiftRes.numberOfResourcesAvailable,
			scheduleName <- shiftRes.name
		)
}

lazyrule ResourceRefrence2PoolResource {
	from
		poolRes : Simul8!ResourcePool
```

```
        to
                ResReffr : Arena!ResourceReference (
                        quantity <- poolRes.priority
                )
}

rule ResourceAction2Action {
        from
                resourecAction : Simul8!ResourceActions
        to
                action : Arena!Action (
                )
}

lazyrule holdResource2DelayRelease {
        from
                holdResource : Simul8!HoldResources
        to
                delayRelease : Arena!DelayRelease (
                )
}

lazyrule ReleaseResource2siezeDelayRelease {
        from
                releaseResource : Simul8!ReleaseResources
        to
                siezeDelayRelease : Arena!SiezeDelayRelease (
                        priority <- releaseResource.priority
                )
}

rule DelayResource2siezeDelay {
        from
                holdResource : Simul8!DelayResources
        to
                siezeDelay : Arena!SiezeDelay (
                        priority <- holdResource.priority
                )
}

rule Hold2Delay {
        from
                holdRes : Simul8!Hold
        to
                delayRes : Arena!Delay (
                )
}
```

APPENDIX 6: ARENA to SIMUL8 Transformation Mapping Rules

ARENA to SIMUL8 Transformation Rules Results.

```
-- @path Simul8=/model.simul8/model/simul8.ecore
-- @path Arena=/model.arena/model/arena.ecore
-- @nsURI Arena= http://model.arena/1.0
-- @nsURI Simul8= http://model.simul8/1.0

module ArenaToSimul8;
create OUT : Simul8 from IN : Arena;

rule Create2SWorkEntryPoint {
	from
		createA : Arena!Create
	to
		workEntryPoint : Simul8!WorkEntryPoint (
			name <- createA.name,
			firstAtStartTimes <- createA.entityPerArrival,
			unlimitedArrivals <- createA.maxArrivals,
			timeUnit <- createA.timeUnit,
			distribution <- createA.timeBetweenArrivals,
			routingOut <- createA.queues->collect(q|thisModule.Queue2StorageBin(q))
		)
}

rule Procees2WorkCenter {
	from
		process : Arena!Process
	to
		workCenter : Simul8!WorkCenter (
			name <- process.name,
			highVolume <- process.reportStatistics,
			distribution <- process.delayType,
			routingIn <- process.inputQueues->collect(q|thisModule.Queue2StorageBin(q)),
			routingOut <- process.ouputQueues->collect(q|thisModule.Queue2Outputbuffer(q))

		)
}


rule Resource2Resource {
	from
		resourceA : Arena!Resources
	to
```

```
		resource : Simul8!Resource (
			name <- resourceA.name,
			cost <- resourceA.busyHour,
			time <- resourceA.idleHour,
			usage <- resourceA.perUse,
			shiftDependence <- resourceA.type,
			utilization <- resourceA.reportStatistic
		)
}

rule Dispose2ExistPoint {
	from
		dispose : Arena!Dispose
	to
		WorkExistPoint : Simul8!WorkExistPoint (
			name <- dispose.name,
			workCompleted <- dispose.workCompleted,
			collectResult <- dispose.collectResult,
			segregateResult <- dispose.recordStatistics,
			haltModelAtLimit <- dispose.haltSimulation,
			highvolume <- dispose.terminateRun
		)
}

rule RunSetUp2SimulationClock {
	from
		runSetup : Arena!RunSetup
	to
		simulClock : Simul8!SimulationClock (
			run <- runSetup.simulate,
			resetToSetup <- runSetup.pauseAfterWarning,
			beepOnCompletion <- runSetup.terminationCondition,
			warmUpPeriod <- runSetup.warmingUpPeriod,
			resultCollectionPeriod <- runSetup.resultCollectionPeriod
		)
}

rule AreKpi2simKpid {
	from
		areKpi : Arena!Kpi
	to
		simKpi : Simul8!Kpid (
			CompletedProductCount <- areKpi.CompletedProductCount,
			cycleTime <- areKpi.cycleTime,
			equipmentUtilization <- areKpi.equipmentUtilization,
			leadTime <- areKpi.leadTime,
			workInProgress <- areKpi.workInProgress
		)
}
```

```
rule ModelToModel {
	from
		ArenModel : Arena!Model
	to
		SmilModel : Simul8!Model (
			name <-  ArenModel.name,
			description <- ArenModel.description,
			timeUnits <- ArenModel.timeUnit
		)
}

lazyrule RunningTimeRunSetUp2RunTimeSimulationClock {
	from
	     --timerunSetUp
:Arena!RunningTime(timerunSetUp.oclIsKindOf(Arena!runSetUp))
		timerunSetUp : Arena!RunningTime
	to
		runTimeSiml : Simul8!RunningTime (
			startTimeAtEachDay <- timerunSetUp.startDayAndTime,
			timeInEachDay <- timerunSetUp.hourPerDay
		)
}

lazyrule RunSpeedRunSetUp2SimulationSpeedClock {
	from
	    --                                                    RunSpeedSetUp
:Arena!SimulationSpeed(RunSpeedSetUp.oclIsKindOf(Arena!runSetUp))
		RunSpeedSetUp : Arena!SimulationSpeed
	to
		simulSpeed : Simul8!RunSpeed (
			speedControl <- RunSpeedSetUp.numberOfReplication,
			typicaMinutesPerRealSecond <- RunSpeedSetUp.replicationLenght
		)
}

lazyrule ClockPropertiesRunSetUp2ClockPropertiesClock {
	from
	     --ClkPperties
:Arena!ClockProperties(ClkPperties.oclIsKindOf(Arena!runSetUp))
		ClkPperties : Arena!ClockProperties
	to
		ClckPperties : Simul8!ClockProperties (
			TimeUnit <- ClkPperties.timeUnit
		)
}

lazyrule Queue2StorageBin {
	from
		 queue : Arena!Queue
	to
```

```
		storage : Simul8!OutputBuffer (
			isHighVolume <- queue.reportStatistics
		)
}

lazyrule Queu2ImputBuffer {
	from
		queue :Arena!Queue
	to
		storage : Simul8!InputBuffer (
			isHighVolume <- queue.reportStatistics
		)
}

lazyrule Queue2Outputbuffer {
	from
		queue : Arena!Queue
	to
		storage : Simul8!StorageBin (
			name <- queue.name,
			isHighVolume <- queue.reportStatistics
		)
}

lazyrule TriangularArrivalType2TriangularDistribution {
	from
	    --arivalTri : Arena!Triangular(arivalTri.oclIsKindOf(Arena!ArrivalType))
		arivalTri : Arena!Triangular
	to
		triDist : Simul8!Triangular (
			mode <- arivalTri.allocation,
			upper <- arivalTri.maximum,
			lower <- arivalTri.minimum
		)
}

lazyrule FixedCapacityArrivalType2FixedDistribution {
	from
	    --arrivalFix : Arena!FixedCapacitry(arrivalFix.oclIsKindOf(Arena!ArrivalType))
		arrivalFix : Arena!FixedCapacitry
	to
		fixDist : Simul8!Fixed (
			value <- arrivalFix.capacity
		)
}

lazyrule UniformArrivalType2RoundedUniformDistribution {
	from

		arivalUni : Arena!Uniform
```

```
	to
		formDist : Simul8!RoundedUniform (
			lowerBound <- arivalUni.minimum,
			upperBound <-  arivalUni.maximum
		)
}


lazyrule RandomArrivalType2ExponentialDistribution {
	from
	      --ramdomarrival:
Arena!Random(ramdomarrival.oclIsKindOf(Arena!ArrivalType))
		ramdomarrival: Arena!Random
	to
		expoDis : Simul8!Exponential (
			vaule <-  ramdomarrival.value,
			average <- ramdomarrival.firstCreation
		)
}

lazyrule NormArrivalType2LogNormDistribution {
	from
	       --normArrival: Arena!Normal(normArrival.oclIsKindOf(Arena!ArrivalType))
		normArrival : Arena!Normal
	to
		LogMormDist : Simul8!LogNormal (
			value <- normArrival.value,
			stdDeviation <- normArrival.stdDeviation,
			unit <- normArrival.units,
			average <- normArrival.allocation
		)
}

lazyrule ShiftResource2ScheduleRulelResourceType {
	from
		schedRes : Arena!ScheduleRes
	to
		shiftRes:  Simul8!Shift  (
			name <- schedRes.scheduleName,
			behaviour <- schedRes.RULE,
			numberOfResourcesAvailable <- schedRes.capacity

		)
}

lazyrule PoolResource2ResourceRefrence {
	from
		resReffr : Arena!ResourceReference
	to
		poolRes : Simul8!ResourcePool (
```

```
				priority <- resReffr.quantity

			)
}

rule Action2ResourceAction{
		from
			action : Arena!Action
		to
			resourecAction : Simul8!ResourceActions (
			)
}

lazyrule DelayRelease2holdResource {
		from
			delayRelease : Arena!DelayRelease
		to
			holdResource: Simul8!HoldResources (
			)
}

lazyrule siezeDelayRelease2ReleaseResource {
		from
			releaseResource : Arena!SiezeDelayRelease
		to
			siezeDelayRelease : Simul8!ReleaseResources (
				priority <- siezeDelayRelease.priority
			)
}

lazyrule siezeDelay2DelayResource {
		from
			siezeDelay : Arena!SiezeDelay
		to
			holdResource : Simul8!DelayResources (
				priority <- siezeDelay.priority
			)
}

lazyrule Delay2Hold {
		from
			delayRes : Arena!Delay
		to
			holdRes : Simul8!Hold (
			)
}
```

APPENDIX 7 ARENA Import File

ProjectParameters												
Project Title	Analyst Name	ProjectDescription	CostingStatistics	QueueStatistics	Transporter Statistics	EntityStatistics	ConveyorStatistics	ProcessStatistics	ResourceStatistics	StationStatistics	ActivityArea Statistics	TankStatistics
Unnamed Project	JYUSUF		0	-1	0	-1	0	0	-1	0	0	0

BasicProcess|Create														
SerialNumber	ModelLevelID	X	Y	UserDescription	Name	Max Batches	Interarrival Type	Schedule	Expression	Value	First Creation	Units	Batch Size	Entity Type
16	1	551	1079		Phone Parts	Infinite	Constant	Schedule 1	1	1	0.0	Hours	1	Entity 1

BasicProcess|Process																		
SerialNumber	ModelLevelID	X	Y	UserDescription	Name	ReportStatistics	Type	Action	ValueAdded	DelayType	Units	Priority	Expression	StDev	Max	Min	Value
33	1	1584	1029		Phone Assembly	Yes	Standard	SD	VA	Constant	Hours	2	1	.2	1.5	.5	1
90	1	1634	2079		Case Packing	Yes	Standard	SDR	VA	Constant	Hours	2	1	.2	1.5	.5	1

BasicProcess|Resource																				
SerialNumber	ModelLevelID	X	Y	UserDescription	Name	ReportStatistics	Usage	Busy	Type	Idle	ScheduleRule	Schedule	Capacity	StatesetN	InitState	FDMName	FDMId	ArenaImportedName	BaseEfficiency	EfficiencySchedule
137	1	0	0		Assembler	Yes	0.0	0.0	Capacity	0.0	Wait	1	1	Phone Assembly States			0		1.0	
138	1	0	0		Packer	Yes	0.0	0.0	Capacity	0.0	Wait	1	1	Case Packing States			0		1.0	

BasicProcess|Queue									
SerialNumber	ModelLevelID	X	Y	UserDescription	Name	ReportStatistics	Type	Attribute	Shared
140	1	0	0		Phone Assembly.Queue	Yes	FIFO	Attribute 1	No
63	1	0	0		Phone Assembly Blocked.Queue	Yes	FIFO	Attribute 1	No
92	1	0	0		Case Packing.Queue	Yes	FIFO	Attribute 1	No

BasicProcess|Dispose						
SerialNumber	ModelLevelID	X	Y	UserDescription	Name	EntStats
117	1	3750	2000		Warehouse	Yes

BasicProcess|Entity													
SerialNumber	ModelLevelID	X	Y	UserDescription	Name	ReportStatistics	Other	InitTranCost	Waiting	Holding Cost	InitNVACost	InitVACost	Picture
17	1	0	0		Entity 1	Yes	0.0	0.0	0.0	0.0	0.0	0.0	Picture.Widgets

BasicProcess|Assign|Assignments										
ModuleSerialNumber	Assignments|Index	Type	AName	OtherName	TypeName	PicName	Column	VName	Value	Row
106	1	Entity Picture	Attribute 1	J	Entity 1	Picture.Box	1	Variable 1	1	1
49	1	Entity Picture	Attribute 1	J	Entity 1	Picture.Telephone	1	Variable 1	1	1

APPENDIX 8: SIMUL8 Import File


```
<SIMUL8XML>
  <SimulationParameters>
    <Trial>
      <Title></Title>
      <Runs>5</Runs>
      <WorkItemType Name="Main Work Item Type" ID="1">
      </WorkItemType>
    </WorkTypes>
    <SimulationObject Name="Phone Part" Type="Work Entry Point" ID="1">
      <DisplayData>
        <Displaytype>4</Displaytype>
           <S8TheType>1000</S8TheType>
          <S8ObjectType>1</S8ObjectType>
          <ObjectID>2</ObjectID>
          <Requnits>1</Requnits>
      <Collectresults>Yes</Collectresults>
      <RouteRNSubStream>5</RouteRNSubStream>
      <IgnoreBlockedRoutes>Yes</IgnoreBlockedRoutes>
      <InterArrivalTimeSampleData>
        <Userates>No</Userates>
        <DistParam1>10</DistParam1>
        <DistParam2>0</DistParam2>
        <DistParam3>0</DistParam3>
        <DistParam4>0</DistParam4>
        <DistribType>7</DistribType>
        <RNSubStream>1</RNSubStream>
        <ReferencedDistribution>0</ReferencedDistribution>
      </InterArrivalTimeSampleData>
        <ReferencedDistribution>0</ReferencedDistribution>
      </batchsizeoutSampleData>
      <flowtimeSampleData>
        <Userates>No</Userates>
        <DistParam1>1000</DistParam1>
        <DistParam2>25</DistParam2>
        <DistParam3>0</DistParam3>
        <DistParam4>0</DistParam4>
        <DistribType>2</DistribType>
        <RNSubStream>2</RNSubStream>
        <ReferencedDistribution>0</ReferencedDistribution>
      </flowtimeSampleData>
      <gaptimeSampleData>
        <Userates>No</Userates>
        <DistParam1>0</DistParam1>
        <DistParam2>0</DistParam2>
        <DistParam3>0</DistParam3>
        <DistParam4>0</DistParam4>
        <DistribType>2</DistribType>
        <RNSubStream>3</RNSubStream>
```

```
      <ReferencedDistribution>0</ReferencedDistribution>
    </gaptimeSampleData>
    <ExitWorkType>1</ExitWorkType>
    <LogicRNSubStream>0</LogicRNSubStream>
    <Finance>
      <CapitalCost>10</CapitalCost>
      <UnitCost>1</UnitCost>
      <TimeCost>0</TimeCost>
      <OtherCost>0</OtherCost>
      <OtherRevenue>0</OtherRevenue>
    </Finance>
    <FirstatZero>No</FirstatZero>
    <Unlimited>Yes</Unlimited>
  </SimulationObject>
  <SimulationObject Name="Phone Assembly" Type="Work Center" ID="2">
    <Index>2</Index>
    <Window>1</Window>
    <DisplayData>
      <Displaytype>4</Displaytype>
    </DisplayData>
    <InputList>
      <Link>
        <S8TheType>1000</S8TheType>
        <S8ObjectType>1</S8ObjectType>
        <ObjectID>1</ObjectID>
        <Requnits>1</Requnits>
      </Link>
    </InputList>
    <OutputList>
      <Link>
        <S8TheType>1000</S8TheType>
        <S8ObjectType>1</S8ObjectType>
        <ObjectID>4</ObjectID>
        <Requnits>1</Requnits>
      </Link>
    </OutputList>
    <Showmylinks>Yes</Showmylinks>
    <MaxConts>1</MaxConts>
    <Collectresults>Yes</Collectresults>
    <Priority>50</Priority>
    <Relresources>Yes</Relresources>
    <RouteRNSubStream>9</RouteRNSubStream>
    <InputRequiredOnOutput>No</InputRequiredOnOutput>
    <IgnoreBlockedRoutes>Yes</IgnoreBlockedRoutes>
    <IgnoreStarvedRoutes>Yes</IgnoreStarvedRoutes>
    <Preference_route>Yes</Preference_route>
    <Routemode>1</Routemode>
    <InRoutemode>4</InRoutemode>
    <Young_Old_UseQueueTime>No</Young_Old_UseQueueTime>
    <Collect_wait_all>No</Collect_wait_all>
```

<Resourcesfirst>Yes</Resourcesfirst>
<Maxattbat>10</Maxattbat>
<Minattbat>1</Minattbat>
<HighVol>No</HighVol>
<HVbatch>No</HVbatch>
<Attbat>0</Attbat>
<Prod_type_att>0</Prod_type_att>
<S8flags>0</S8flags>
<Fixed_prod_type>0</Fixed_prod_type>
<TimingStyle>0</TimingStyle>
  <Collect_assemble>Yes</Collect_assemble>
<OperationTimeSampleData>
  <Userates>No</Userates>
  <DistParam1>10</DistParam1>
  <DistParam2>2.5</DistParam2>
  <DistParam3>0</DistParam3>
  <DistParam4>0</DistParam4>
  <DistribType>1</DistribType>
  <RNSubStream>6</RNSubStream>
  <ReferencedDistribution>0</ReferencedDistribution>
</OperationTimeSampleData>
<flowtimeSampleData>
  <Userates>No</Userates>
  <DistParam1>1000</DistParam1>
  <DistParam2>25</DistParam2>
  <DistParam3>0</DistParam3>
  <DistParam4>0</DistParam4>
  <DistribType>2</DistribType>
  <RNSubStream>7</RNSubStream>
  <ReferencedDistribution>0</ReferencedDistribution>
</flowtimeSampleData>
<gaptimeSampleData>
  <Userates>No</Userates>
  <DistParam1>0</DistParam1>
  <DistParam2>0</DistParam2>
  <DistParam3>0</DistParam3>
  <DistParam4>0</DistParam4>
  <DistribType>2</DistribType>
  <RNSubStream>8</RNSubStream>
  <ReferencedDistribution>0</ReferencedDistribution>
</gaptimeSampleData>
<batchsizeoutSampleData>
  <Userates>No</Userates>
  <DistParam1>1</DistParam1>
  <DistParam2>0</DistParam2>
  <DistParam3>0</DistParam3>
  <DistParam4>0</DistParam4>
  <DistribType>2</DistribType>
  <RNSubStream>10</RNSubStream>
  <ReferencedDistribution>0</ReferencedDistribution>

</batchsizeoutSampleData>
<LogicRNSubStream>0</LogicRNSubStream>
<ExitWorkType>0</ExitWorkType>
<RouteLabel>0</RouteLabel>
<PriorityLabel>0</PriorityLabel>
<IndexingGroup>0</IndexingGroup>
<Everyresult>No</Everyresult>
<WorkItemImage>0</WorkItemImage>
<InterruptonStorage></InterruptonStorage>
<Finance>
<changeOverSampleData>
<Userates>No</Userates>
<DistParam1>0</DistParam1>
<DistParam2>0</DistParam2>
<DistParam3>0</DistParam3>
<DistParam4>0</DistParam4>
<DistribType>2</DistribType>
<RNSubStream>11</RNSubStream>
<ReferencedDistribution>0</ReferencedDistribution
<Work_time_between_setups>0</Work_time_between_setups>
<Check_exit_clear_routein>No</Check_exit_clear_routein>
<Wait_for_interval>No</Wait_for_interval>
</SimulationObject>
<SimulationObject Name="Queue for Release Assembly" Type="Storage Area" ID="3">
<Displaytype>3</Displaytype>
<S8TheType>1000</S8TheType>
<S8ObjectType>1</S8ObjectType>
<ObjectID>5</ObjectID>
<Requnits>1</Requnits>
<Transitlink>1</Transitlink>
</Link>
</InputList>
<OutputList>
<Link>
<S8TheType>1000</S8TheType>
<S8ObjectType>1</S8ObjectType>
<ObjectID>6</ObjectID>
<Requnits>1</Requnits>
<Transitlink>2</Transitlink>
</Link>
<ExitWorkType>1</ExitWorkType>
<PriorityLabel>0</PriorityLabel>
<ShelfLifeLabel>0</ShelfLifeLabel>
<Include_label_display>No</Include_label_display>
<HighVol>No</HighVol>
<LIFO>No</LIFO>
<Everyresult>No</Everyresult>
<Expiretime>-1</Expiretime>
<ResultsSegLabel>0</ResultsSegLabel>
<Finance>

```
<SimulationObject Name="Assign Phone" Type="Work Center" ID="4">
   <Displaytype>4</Displaytype>
    <S8TheType>1000</S8TheType>
     <S8ObjectType>1</S8ObjectType>
     <ObjectID>2</ObjectID>
     <Requnits>1</Requnits>
   </Link>
  </InputList>
  <OutputList>
   <Link>
     <S8TheType>1000</S8TheType>
     <S8ObjectType>1</S8ObjectType>
     <ObjectID>5</ObjectID>
     <Requnits>1</Requnits>
   </Link>
  </OutputList>
  <Showmylinks>Yes</Showmylinks>
  <MaxConts>1</MaxConts>
  <Collectresults>Yes</Collectresults>
  <Priority>50</Priority>
  <Relresources>Yes</Relresources>
  <RouteRNSubStream>15</RouteRNSubStream>
  <InputRequiredOnOutput>No</InputRequiredOnOutput>
  <IgnoreBlockedRoutes>Yes</IgnoreBlockedRoutes>
  <IgnoreStarvedRoutes>Yes</IgnoreStarvedRoutes>
  <Preference_route>Yes</Preference_route>
  <Routemode>1</Routemode>
  <InRoutemode>4</InRoutemode>
  <Young_Old_UseQueueTime>No</Young_Old_UseQueueTime>
  <Collect_wait_all>No</Collect_wait_all>
  <Resourcesfirst>Yes</Resourcesfirst>
  <Maxattbat>10</Maxattbat>
  <Minattbat>1</Minattbat>
  <HighVol>No</HighVol>
  <HVbatch>No</HVbatch>
  <Matchatt>0</Matchatt>
  <Attbat>0</Attbat>
  <Prod_type_att>0</Prod_type_att>
  <S8flags>0</S8flags>
  <Fixed_prod_type>0</Fixed_prod_type>
  <TimingStyle>0</TimingStyle>
  <TISmode>0</TISmode>
  <Collect_assemble>Yes</Collect_assemble>
  <OperationTimeSampleData>
   <Userates>No</Userates>
   <DistParam1>10</DistParam1>
   <DistParam2>2.5</DistParam2>
   <DistParam3>0</DistParam3>
   <DistParam4>0</DistParam4>
   <DistribType>1</DistribType>
```

```
    <RNSubStream>12</RNSubStream>
    <ReferencedDistribution>0</ReferencedDistribution>
  </OperationTimeSampleData>
  <flowtimeSampleData>
    <Userates>No</Userates>
    <DistParam1>1000</DistParam1>
    <DistParam2>25</DistParam2>
    <DistParam3>0</DistParam3>
    <DistParam4>0</DistParam4>
    <DistribType>2</DistribType>
    <RNSubStream>13</RNSubStream>
    <ReferencedDistribution>0</ReferencedDistribution>
  </flowtimeSampleData>
  <gaptimeSampleData>
    <Userates>No</Userates>
    <DistParam1>0</DistParam1>
    <DistParam2>0</DistParam2>
    <DistParam3>0</DistParam3>
    <DistParam4>0</DistParam4>
    <DistribType>2</DistribType>
    <RNSubStream>14</RNSubStream>
    <ReferencedDistribution>0</ReferencedDistribution>
  </gaptimeSampleData>
  <batchsizeoutSampleData>
    <Userates>No</Userates>
    <DistParam1>1</DistParam1>
    <DistParam2>0</DistParam2>
    <DistParam3>0</DistParam3>
    <DistParam4>0</DistParam4>
    <DistribType>2</DistribType>
    <RNSubStream>16</RNSubStream>
    <ReferencedDistribution>0</ReferencedDistribution>
  </batchsizeoutSampleData>
  <LogicRNSubStream>0</LogicRNSubStream>
  <ExitWorkType>0</ExitWorkType>
  <RouteLabel>0</RouteLabel>
  <PriorityLabel>0</PriorityLabel>
  <IndexingGroup>0</IndexingGroup>
  <Everyresult>No</Everyresult>
  <WorkItemImage>0</WorkItemImage>
  <InterruptonStorage></InterruptonStorage>
  <Finance>
    <CapitalCost>10</CapitalCost>
    <UnitCost>1</UnitCost>
    <TimeCost>0</TimeCost>
    <OtherCost>0</OtherCost>
    <OtherRevenue>0</OtherRevenue>
  </Finance>
  <changeOverSampleData>
    <Userates>No</Userates>
```

<DistParam1>0</DistParam1>
<DistParam2>0</DistParam2>
<DistParam3>0</DistParam3>
<DistParam4>0</DistParam4>
<DistribType>2</DistribType>
<RNSubStream>17</RNSubStream>
<ReferencedDistribution>0</ReferencedDistribution>
</changeOverSampleData>
<ChangeOverLabel>0</ChangeOverLabel>
<ChangeOverStyle>0</ChangeOverStyle>
<Work_time_between_setups>0</Work_time_between_setups>
<Check_exit_clear_routein>No</Check_exit_clear_routein>
<Wait_for_interval>No</Wait_for_interval>
</SimulationObject>
<SimulationObject Name="Phone Assembly Blocked" Type="Work Center" ID="5">
<Index>5</Index>
<Window>1</Window>
<DisplayData>
<Displaytype>4</Displaytype>
<Link>
<S8TheType>1000</S8TheType>
<S8ObjectType>1</S8ObjectType>
<ObjectID>4</ObjectID>
<Requnits>1</Requnits>
</Link>
</InputList>
<OutputList>
<Link>
<S8TheType>1000</S8TheType>
<S8ObjectType>1</S8ObjectType>
<ObjectID>3</ObjectID>
<Requnits>1</Requnits>
<Transitlink>1</Transitlink>
</Link>
</OutputList>
<Showmylinks>Yes</Showmylinks>
<MaxConts>1</MaxConts>
<Collectresults>Yes</Collectresults>
<Priority>50</Priority>
<Relresources>Yes</Relresources>
<RouteRNSubStream>21</RouteRNSubStream>
<InputRequiredOnOutput>No</InputRequiredOnOutput>
<IgnoreBlockedRoutes>Yes</IgnoreBlockedRoutes>
<IgnoreStarvedRoutes>Yes</IgnoreStarvedRoutes>
<Preference_route>Yes</Preference_route>
<Routemode>1</Routemode>
<InRoutemode>4</InRoutemode>
<Young_Old_UseQueueTime>No</Young_Old_UseQueueTime>
<Collect_wait_all>No</Collect_wait_all>
<Resourcesfirst>Yes</Resourcesfirst>

```
<Maxattbat>10</Maxattbat>
<Minattbat>1</Minattbat>
<HighVol>No</HighVol>
<HVbatch>No</HVbatch>
<Matchatt>0</Matchatt>
<Attbat>0</Attbat>
<Prod_type_att>0</Prod_type_att>
<S8flags>0</S8flags>
<Fixed_prod_type>0</Fixed_prod_type>
<TimingStyle>4</TimingStyle>
<TISmode>0</TISmode>
<Collect_assemble>Yes</Collect_assemble>
<OperationTimeSampleData>
  <Userates>No</Userates>
  <DistParam1>10</DistParam1>
  <DistParam2>0</DistParam2>
  <DistParam3>0</DistParam3>
  <DistParam4>0</DistParam4>
  <DistribType>2</DistribType>
  <RNSubStream>18</RNSubStream>
  <ReferencedDistribution>0</ReferencedDistribution>
</OperationTimeSampleData>
<flowtimeSampleData>
  <Userates>No</Userates>
  <DistParam1>1000</DistParam1>
  <DistParam2>25</DistParam2>
  <DistParam3>0</DistParam3>
  <DistParam4>0</DistParam4>
  <DistribType>2</DistribType>
  <RNSubStream>19</RNSubStream>
  <ReferencedDistribution>0</ReferencedDistribution>
</flowtimeSampleData>
<gaptimeSampleData>
  <Userates>No</Userates>
  <DistParam1>0</DistParam1>
  <DistParam2>0</DistParam2>
  <DistParam3>0</DistParam3>
  <DistParam4>0</DistParam4>
  <DistribType>2</DistribType>
  <RNSubStream>20</RNSubStream>
  <ReferencedDistribution>0</ReferencedDistribution>
</gaptimeSampleData>
<batchsizeoutSampleData>
  <Userates>No</Userates>
  <DistParam1>1</DistParam1>
  <DistParam2>0</DistParam2>
  <DistParam3>0</DistParam3>
  <DistParam4>0</DistParam4>
  <DistribType>2</DistribType>
  <RNSubStream>22</RNSubStream>
```

```
    <ReferencedDistribution>0</ReferencedDistribution>
  </batchsizeoutSampleData>
  <LogicRNSubStream>0</LogicRNSubStream>
  <ExitWorkType>0</ExitWorkType>
  <RouteLabel>0</RouteLabel>
  <PriorityLabel>0</PriorityLabel>
  <IndexingGroup>0</IndexingGroup>
  <Everyresult>No</Everyresult>
  <WorkItemImage>0</WorkItemImage>
  <InterruptonStorage></InterruptonStorage>
  <Finance>
    <CapitalCost>10</CapitalCost>
    <UnitCost>1</UnitCost>
    <TimeCost>0</TimeCost>
    <OtherCost>0</OtherCost>
    <OtherRevenue>0</OtherRevenue>
  </Finance>
  <changeOverSampleData>
    <Userates>No</Userates>
    <DistParam1>0</DistParam1>
    <DistParam2>0</DistParam2>
    <DistParam3>0</DistParam3>
    <DistParam4>0</DistParam4>
    <DistribType>2</DistribType>
    <RNSubStream>23</RNSubStream>
    <ReferencedDistribution>0</ReferencedDistribution>
  </changeOverSampleData>
  <ChangeOverLabel>0</ChangeOverLabel>
  <ChangeOverStyle>0</ChangeOverStyle>
  <Work_time_between_setups>0</Work_time_between_setups>
  <Check_exit_clear_routein>No</Check_exit_clear_routein>
  <Wait_for_interval>Yes</Wait_for_interval>
</SimulationObject>
<SimulationObject Name="Release Assembly" Type="Work Center" ID="6">
  <Index>6</Index>
  <Window>1</Window>
  <DisplayData>
    <Displaytype>4</Displaytype>
    </Link>
  </OutputList>
  <Showmylinks>Yes</Showmylinks>
  <MaxConts>1</MaxConts>
  <Collectresults>Yes</Collectresults>
  <Priority>50</Priority>
  <Relresources>Yes</Relresources>
  <RouteRNSubStream>27</RouteRNSubStream>
  <InputRequiredOnOutput>No</InputRequiredOnOutput>
  <IgnoreBlockedRoutes>Yes</IgnoreBlockedRoutes>
  <IgnoreStarvedRoutes>Yes</IgnoreStarvedRoutes>
  <Preference_route>Yes</Preference_route>
```

<Routemode>1</Routemode>
<InRoutemode>4</InRoutemode>
<Young_Old_UseQueueTime>No</Young_Old_UseQueueTime>
<Collect_wait_all>No</Collect_wait_all>
<Resourcesfirst>Yes</Resourcesfirst>
<Maxattbat>10</Maxattbat>
<Minattbat>1</Minattbat>
<HighVol>No</HighVol>
<HVbatch>No</HVbatch>
<Matchatt>0</Matchatt>
<Attbat>0</Attbat>
<Prod_type_att>0</Prod_type_att>
<S8flags>0</S8flags>
<Fixed_prod_type>0</Fixed_prod_type>
<TimingStyle>0</TimingStyle>
<TISmode>0</TISmode>
<Collect_assemble>Yes</Collect_assemble>
<OperationTimeSampleData>
  <Userates>No</Userates>
  <DistParam1>10</DistParam1>
  <DistParam2>2.5</DistParam2>
  <DistParam3>0</DistParam3>
  <DistParam4>0</DistParam4>
  <DistribType>1</DistribType>
  <RNSubStream>24</RNSubStream>
  <ReferencedDistribution>0</ReferencedDistribution>
</OperationTimeSampleData>
<flowtimeSampleData>
  <Userates>No</Userates>
  <DistParam1>1000</DistParam1>
  <DistParam2>25</DistParam2>
  <DistParam3>0</DistParam3>
  <DistParam4>0</DistParam4>
  <DistribType>2</DistribType>
  <RNSubStream>25</RNSubStream>
  <ReferencedDistribution>0</ReferencedDistribution>
</flowtimeSampleData>
<gaptimeSampleData>
  <Userates>No</Userates>
  <DistParam1>0</DistParam1>
  <DistParam2>0</DistParam2>
  <DistParam3>0</DistParam3>
  <DistParam4>0</DistParam4>
  <DistribType>2</DistribType>
  <RNSubStream>26</RNSubStream>
  <ReferencedDistribution>0</ReferencedDistribution>
</gaptimeSampleData>
<batchsizeoutSampleData>
  <Userates>No</Userates>
  <DistParam1>1</DistParam1>

```
    <DistParam2>0</DistParam2>
    <DistParam3>0</DistParam3>
    <DistParam4>0</DistParam4>
    <DistribType>2</DistribType>
    <RNSubStream>28</RNSubStream>
    <ReferencedDistribution>0</ReferencedDistribution>
  </batchsizeoutSampleData>
  <LogicRNSubStream>0</LogicRNSubStream>
  <ExitWorkType>0</ExitWorkType>
  <RouteLabel>0</RouteLabel>
  <PriorityLabel>0</PriorityLabel>
  <IndexingGroup>0</IndexingGroup>
  <Everyresult>No</Everyresult>
  <WorkItemImage>0</WorkItemImage>
  <InterruptonStorage></InterruptonStorage>
  <Finance>
    <CapitalCost>10</CapitalCost>
    <UnitCost>1</UnitCost>
    <TimeCost>0</TimeCost>
    <OtherCost>0</OtherCost>
    <OtherRevenue>0</OtherRevenue>
  </Finance>
  <changeOverSampleData>
    <Userates>No</Userates>
    <DistParam1>0</DistParam1>
    <DistParam2>0</DistParam2>
    <DistParam3>0</DistParam3>
    <DistParam4>0</DistParam4>
    <DistribType>2</DistribType>
    <RNSubStream>29</RNSubStream>
    <ReferencedDistribution>0</ReferencedDistribution>
  </changeOverSampleData>
  <ChangeOverLabel>0</ChangeOverLabel>
  <ChangeOverStyle>0</ChangeOverStyle>
  <Work_time_between_setups>0</Work_time_between_setups>
  <Check_exit_clear_routein>No</Check_exit_clear_routein>
  <Wait_for_interval>No</Wait_for_interval>
</SimulationObject>
<SimulationObject Name="Case Packing" Type="Work Center" ID="7">
  <Index>7</Index>
  <Window>1</Window>
  <DisplayData>
    <Showworkitem>No</Showworkitem>
    <CmImageOnDisplay>0</CmImageOnDisplay>
    <Color1>6498846</Color1>
    <Color2>15646080</Color2>
    <Orientation>
    </Orientation>
    <ScaleX>1</ScaleX>
    <ScaleY>1</ScaleY>
```

```
  </Link>
</InputList>
<OutputList>
  <Link>
    <S8TheType>1000</S8TheType>
    <S8ObjectType>1</S8ObjectType>
    <ObjectID>8</ObjectID>
    <Requnits>1</Requnits>
  </Link>
</OutputList>
<Showmylinks>Yes</Showmylinks>
<MaxConts>1</MaxConts>
<Collectresults>Yes</Collectresults>
<Priority>50</Priority>
<Relresources>Yes</Relresources>
<RouteRNSubStream>33</RouteRNSubStream>
<InputRequiredOnOutput>No</InputRequiredOnOutput>
<IgnoreBlockedRoutes>Yes</IgnoreBlockedRoutes>
<IgnoreStarvedRoutes>Yes</IgnoreStarvedRoutes>
<Preference_route>Yes</Preference_route>
<Routemode>1</Routemode>
<InRoutemode>4</InRoutemode>
<Young_Old_UseQueueTime>No</Young_Old_UseQueueTime>
<Collect_wait_all>No</Collect_wait_all>
<Resourcesfirst>Yes</Resourcesfirst>
<Maxattbat>10</Maxattbat>
<Minattbat>1</Minattbat>
<HighVol>No</HighVol>
<HVbatch>No</HVbatch>
<Matchatt>0</Matchatt>
<Attbat>0</Attbat>
<Prod_type_att>0</Prod_type_att>
<S8flags>0</S8flags>
<Fixed_prod_type>0</Fixed_prod_type>
<TimingStyle>0</TimingStyle>
<TISmode>0</TISmode>
<Collect_assemble>Yes</Collect_assemble>
<OperationTimeSampleData>
  <Userates>No</Userates>
  <DistParam1>10</DistParam1>
  <DistParam2>2.5</DistParam2>
  <DistParam3>0</DistParam3>
  <DistParam4>0</DistParam4>
  <DistribType>1</DistribType>
  <RNSubStream>30</RNSubStream>
  <ReferencedDistribution>0</ReferencedDistribution>
</OperationTimeSampleData>
<flowtimeSampleData>
  <Userates>No</Userates>
  <DistParam1>1000</DistParam1>
```

```xml
    <DistParam2>25</DistParam2>
    <DistParam3>0</DistParam3>
    <DistParam4>0</DistParam4>
    <DistribType>2</DistribType>
    <RNSubStream>31</RNSubStream>
    <ReferencedDistribution>0</ReferencedDistribution>
  </flowtimeSampleData>
  <gaptimeSampleData>
    <Userates>No</Userates>
    <DistParam1>0</DistParam1>
    <DistParam2>0</DistParam2>
    <DistParam3>0</DistParam3>
    <DistParam4>0</DistParam4>
    <DistribType>2</DistribType>
    <RNSubStream>32</RNSubStream>
    <ReferencedDistribution>0</ReferencedDistribution>
  </gaptimeSampleData>
  <batchsizeoutSampleData>
    <Userates>No</Userates>
    <DistParam1>1</DistParam1>
    <DistParam2>0</DistParam2>
    <DistParam3>0</DistParam3>
    <DistParam4>0</DistParam4>
    <DistribType>2</DistribType>
    <RNSubStream>34</RNSubStream>
    <ReferencedDistribution>0</ReferencedDistribution>
  </batchsizeoutSampleData>
  <LogicRNSubStream>0</LogicRNSubStream>
  <ExitWorkType>0</ExitWorkType>
  <RouteLabel>0</RouteLabel>
  <PriorityLabel>0</PriorityLabel>
  <IndexingGroup>0</IndexingGroup>
  <Everyresult>No</Everyresult>
  <WorkItemImage>0</WorkItemImage>
  <InterruptonStorage></InterruptonStorage>
  <Finance>
    <CapitalCost>10</CapitalCost>
    <UnitCost>1</UnitCost>
    <TimeCost>0</TimeCost>
    <OtherCost>0</OtherCost>
    <OtherRevenue>0</OtherRevenue>
  </Finance>
  <changeOverSampleData>
    <Userates>No</Userates>
    <DistParam1>0</DistParam1>
    <DistParam2>0</DistParam2>
    <DistParam3>0</DistParam3>
    <DistParam4>0</DistParam4>
    <DistribType>2</DistribType>
    <RNSubStream>35</RNSubStream>
```

<ReferencedDistribution>0</ReferencedDistribution>
</changeOverSampleData>
<ChangeOverLabel>0</ChangeOverLabel>
<ChangeOverStyle>0</ChangeOverStyle>
<Work_time_between_setups>0</Work_time_between_setups>
<Check_exit_clear_routein>No</Check_exit_clear_routein>
<Wait_for_interval>No</Wait_for_interval>
</SimulationObject>
<SimulationObject Name="Assign Box Entity" Type="Work Center" ID="8">
<Index>8</Index>
<Window>1</Window>
<DisplayData>
<Displaytype>4</Displaytype>
<X1>418</X1>
<Y1>308</Y1>
<X2>450</X2>
<Y2>340</Y2>
<Xinc>-10</Xinc>
<Yinc>0</Yinc>
<TitleOffsetX>15</TitleOffsetX>
<TitleOffsetY>-24</TitleOffsetY>
<TitleWidth>0</TitleWidth>
<Invisible>No</Invisible>
<Showtitle>Yes</Showtitle>
<Showcount>Yes</Showcount>
<Showimage>Yes</Showimage>
<Showworkitem>No</Showworkitem>
<CmImageOnDisplay>0</CmImageOnDisplay>
<Color1>6498846</Color1>
<Color2>15646080</Color2>
<Orientation>
<Ori_11>1</Ori_11>
<Ori_22>1</Ori_22>
<Ori_33>1</Ori_33>
<Ori_44>1</Ori_44>
</Orientation>
<ScaleX>1</ScaleX>
<ScaleY>1</ScaleY>
<ScaleZ>1</ScaleZ>
<ColorRValue>0.699999999999818</ColorRValue>
<ColorGValue>0.300000000000182</ColorGValue>
<ColorBValue>0.300000000000182</ColorBValue>
</DisplayData>
<InputList>
<Link>
<S8TheType>1000</S8TheType>
<S8ObjectType>1</S8ObjectType>
<ObjectID>7</ObjectID>
<Requnits>1</Requnits>
</Link>

</InputList>
<OutputList>
  <Link>
    <S8TheType>1000</S8TheType>
    <S8ObjectType>1</S8ObjectType>
    <ObjectID>11</ObjectID>
    <Requnits>1</Requnits>
    <Transitlink>3</Transitlink>
  </Link>
</OutputList>
<Showmylinks>Yes</Showmylinks>
<MaxConts>1</MaxConts>
<Collectresults>Yes</Collectresults>
<Priority>50</Priority>
<Relresources>Yes</Relresources>
<RouteRNSubStream>39</RouteRNSubStream>
<InputRequiredOnOutput>No</InputRequiredOnOutput>
<IgnoreBlockedRoutes>Yes</IgnoreBlockedRoutes>
<IgnoreStarvedRoutes>Yes</IgnoreStarvedRoutes>
<Preference_route>Yes</Preference_route>
<Routemode>1</Routemode>
<InRoutemode>4</InRoutemode>
<Young_Old_UseQueueTime>No</Young_Old_UseQueueTime>
<Collect_wait_all>No</Collect_wait_all>
<Resourcesfirst>Yes</Resourcesfirst>
<Maxattbat>10</Maxattbat>
<Minattbat>1</Minattbat>
<HighVol>No</HighVol>
<HVbatch>No</HVbatch>
<Matchatt>0</Matchatt>
<Attbat>0</Attbat>
<Prod_type_att>0</Prod_type_att>
<S8flags>0</S8flags>
<Fixed_prod_type>0</Fixed_prod_type>
<TimingStyle>0</TimingStyle>
<TISmode>0</TISmode>
<Collect_assemble>Yes</Collect_assemble>
<OperationTimeSampleData>
  <Userates>No</Userates>
  <DistParam1>10</DistParam1>
  <DistParam2>2.5</DistParam2>
  <DistParam3>0</DistParam3>
  <DistParam4>0</DistParam4>
  <DistribType>1</DistribType>
  <RNSubStream>36</RNSubStream>
  <ReferencedDistribution>0</ReferencedDistribution>
</OperationTimeSampleData>
<flowtimeSampleData>
  <Userates>No</Userates>
  <DistParam1>1000</DistParam1>

<DistParam2>25</DistParam2>
<DistParam3>0</DistParam3>
<DistParam4>0</DistParam4>
<DistribType>2</DistribType>
<RNSubStream>37</RNSubStream>
<ReferencedDistribution>0</ReferencedDistribution>
</flowtimeSampleData>
<gaptimeSampleData>
<Userates>No</Userates>
<DistParam1>0</DistParam1>
<DistParam2>0</DistParam2>
<DistParam3>0</DistParam3>
<DistParam4>0</DistParam4>
<DistribType>2</DistribType>
<RNSubStream>38</RNSubStream>
<ReferencedDistribution>0</ReferencedDistribution>
</gaptimeSampleData>
<batchsizeoutSampleData>
<Userates>No</Userates>
<DistParam1>1</DistParam1>
<DistParam2>0</DistParam2>
<DistParam3>0</DistParam3>
<DistParam4>0</DistParam4>
<DistribType>2</DistribType>
<RNSubStream>40</RNSubStream>
<ReferencedDistribution>0</ReferencedDistribution>
</batchsizeoutSampleData>
<LogicRNSubStream>0</LogicRNSubStream>
<ExitWorkType>0</ExitWorkType>
<RouteLabel>0</RouteLabel>
<PriorityLabel>0</PriorityLabel>
<IndexingGroup>0</IndexingGroup>
<Everyresult>No</Everyresult>
<WorkItemImage>0</WorkItemImage>
<InterruptonStorage></InterruptonStorage>
<Finance>
<CapitalCost>10</CapitalCost>
<UnitCost>1</UnitCost>
<TimeCost>0</TimeCost>
<OtherCost>0</OtherCost>
<OtherRevenue>0</OtherRevenue>
</Finance>
<changeOverSampleData>
<Userates>No</Userates>
<DistParam1>0</DistParam1>
<DistParam2>0</DistParam2>
<DistParam3>0</DistParam3>
<DistParam4>0</DistParam4>
<DistribType>2</DistribType>
<RNSubStream>41</RNSubStream>

```
    <ReferencedDistribution>0</ReferencedDistribution>
  </changeOverSampleData>
  <ChangeOverLabel>0</ChangeOverLabel>
  <ChangeOverStyle>0</ChangeOverStyle>
  <Work_time_between_setups>0</Work_time_between_setups>
  <Check_exit_clear_routein>No</Check_exit_clear_routein>
  <Wait_for_interval>No</Wait_for_interval>
</SimulationObject>
<SimulationObject Name="Warehouse" Type="Work Complete" ID="11">
  <Index>11</Index>
  <Window>1</Window>
  <DisplayData>
    <Displaytype>4</Displaytype>
```